An Introduction to (Hybrid) Probabilistic Logic Programming

Luc De Raedt

Hybrid Reasoning 2016 Freiburg

DTAI

MACHINE LEARNING

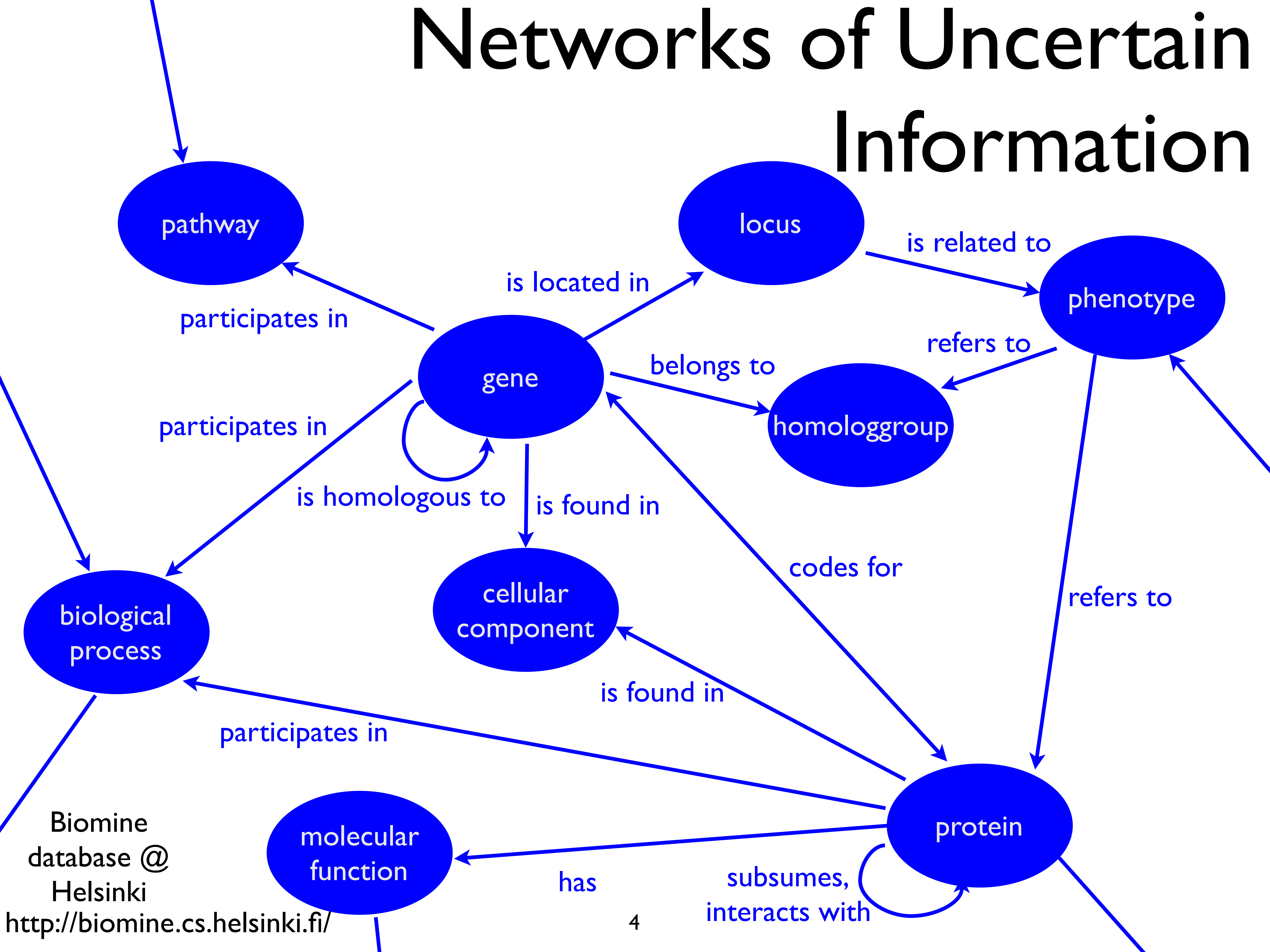
KU LEUVEN

Overview

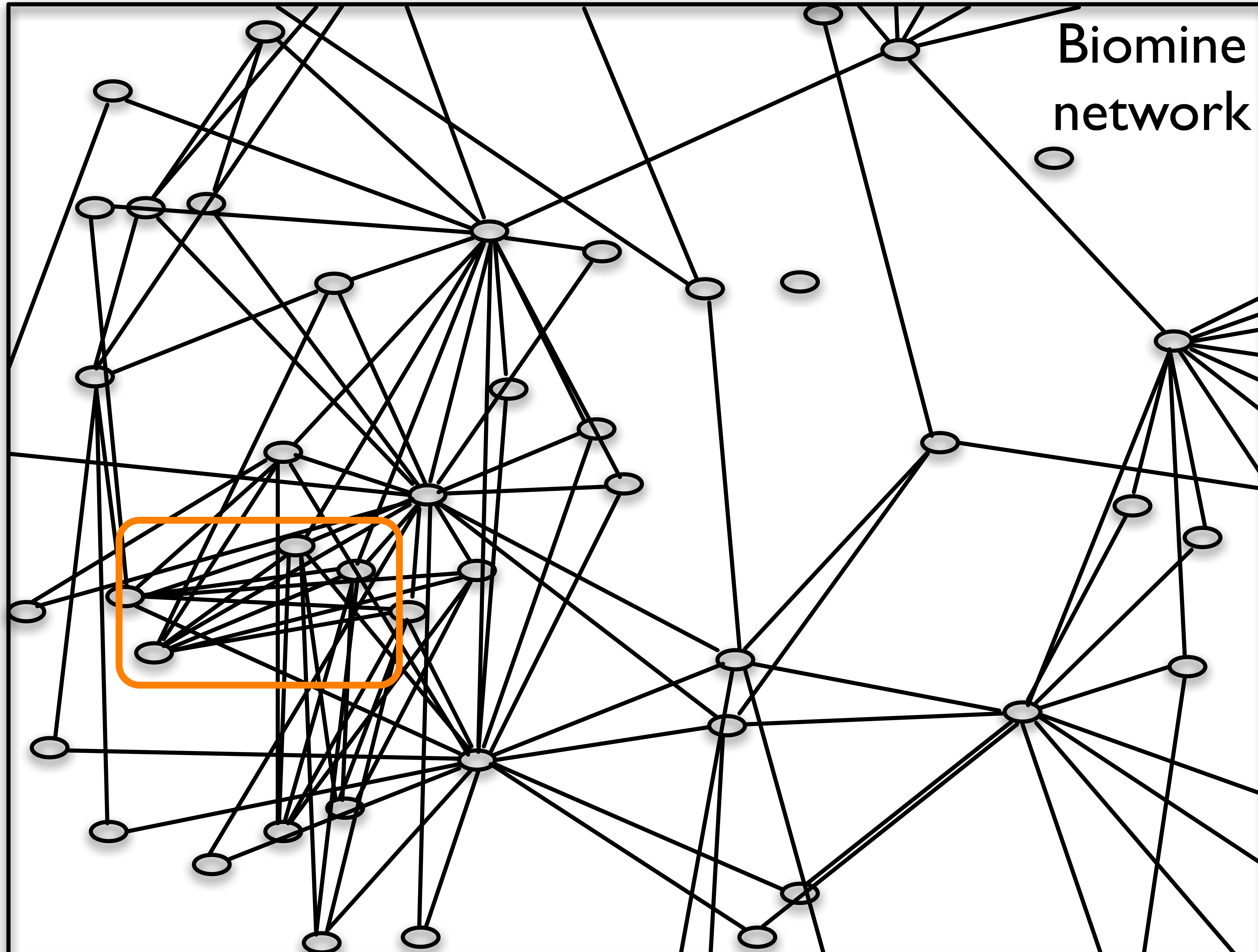
- Part I : An introduction to Prob. Logic Programming and the relation to alternative frameworks
- Part II : Inference
- Part III: Learning
- Part IV: Dynamics & Continuous distributions for Relational Tracking (in Robotics)
- *Focus on ProbLog line of research at KU Leuven*

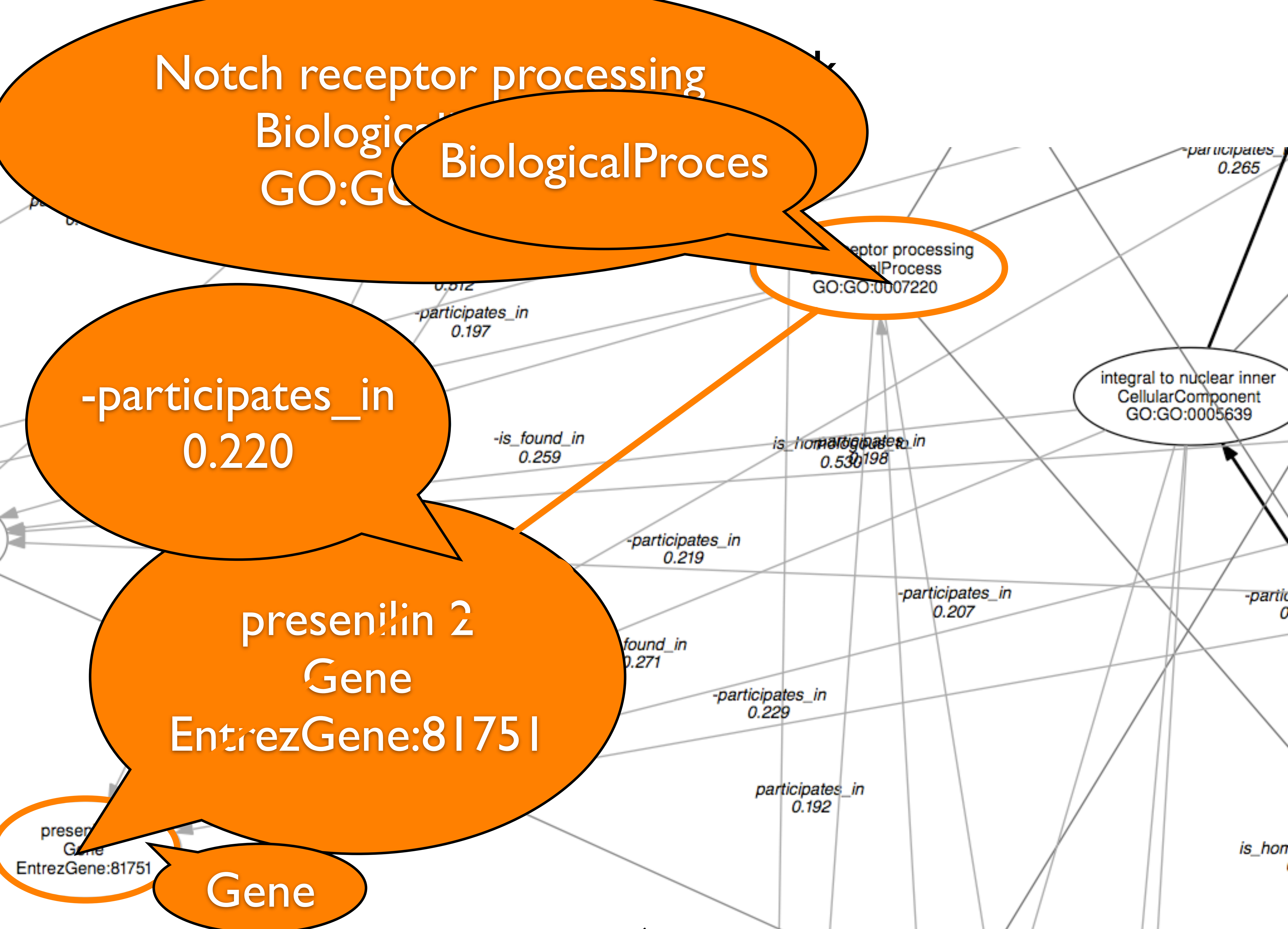
PART 1: Intro to PLP

Networks of Uncertain Information



Biomine database @ Helsinki
http://biomine.cs.helsinki.fi/


Biomine
network


Notch receptor processing
BiologicalProces
-participates_in
0.220
presenilin 2
Gene
EntrezGene:81751
Gene
0.265
-participates_in
0.197
GO:GO:0007220
-is_found_in
0.259
integral to nuclear inner
CellularComponent
GO:GO:0005639
-participates_in
0.219
-participates_in
0.207
-participates_in
0.229
participates_in
0.192
EntrezGene:81751

Phenetic

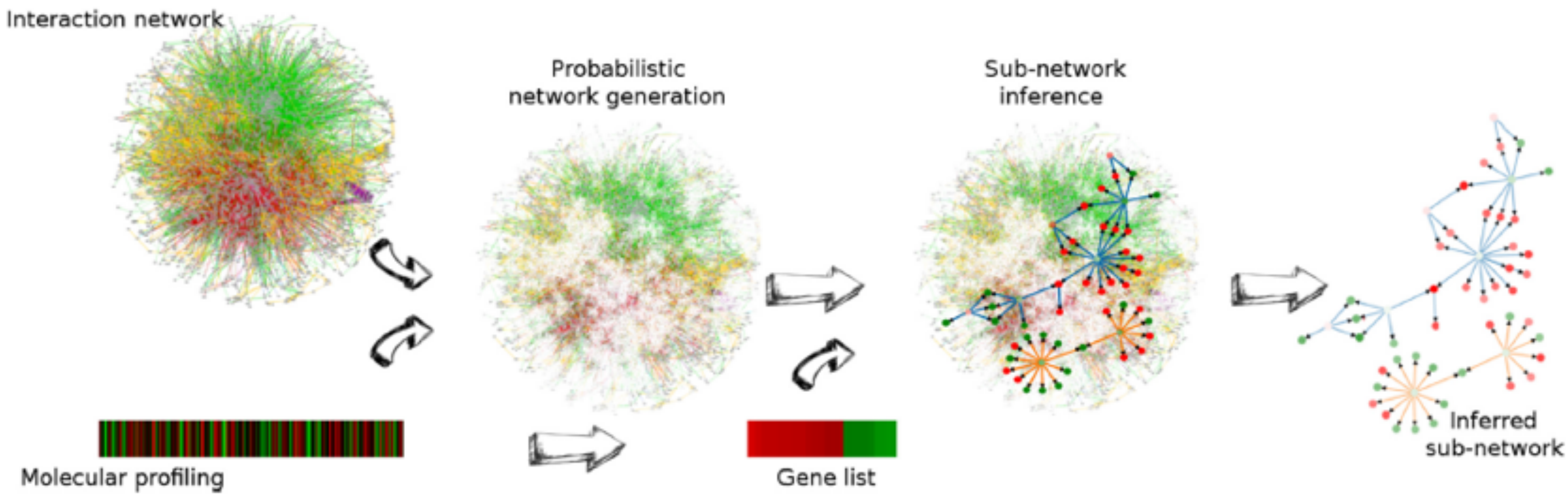


Figure 1. Overview of PheNetic, a web service for network-based interpretation of 'omics' data. The web service uses as input a genome wide interaction network for the organism of interest, a user generated molecular profiling data set and a gene list derived from these data. Interaction networks for a wide variety of organisms are readily available from the web server. Using the uploaded user-generated molecular data the interaction network is converted into a probabilistic network: edges receive a probability proportional to the levels measured for the terminal nodes in the molecular profiling data set. This probabilistic interaction network is used to infer the sub-network that best links the genes from the gene list. The inferred sub-network provides a trade-off between linking as many genes as possible from the gene list and selecting the least number of edges.

- Causes: Mutations
  - All related to similar phenotype
- Effects: Differentially expressed genes
- 27 000 cause effect pairs

- Interaction network:
  - 3063 nodes
    - Genes
    - Proteins
  - 16794 edges
    - Molecular interactions
    - Uncertain

- Goal: connect causes to effects through common subnetwork
  - = Find mechanism
- Techniques:
  - DTProbLog
  - Approximate inference

Can we find the mechanism connecting causes to effects?

[De Maeyer et al., Molecular Biosystems 13, NAR 15]

Example: Information Extraction

Recently-Learned Facts twitter

Refresh

instance	iteration	date learned	confidence
kelly_andrews is a female	826	29-mar-2014	98.7
investment_next_year is an economic sector	829	10-apr-2014	95.3
shibenik is a geopolitical entity that is an organization	829	10-apr-2014	97.2
quality_web_design_work is a character trait	826	29-mar-2014	91.0
mercedes_benz_cls_by_carlsson is an automobile manufacturer	829	10-apr-2014	95.2
social_work is an academic program at the university rutgers_university	827	02-apr-2014	93.8
dante wrote the book the_divine_comedy	826	29-mar-2014	93.8
willie_aames was born in the city los_angeles	831	16-apr-2014	100.0
kitt_peak is a mountain in the state or province arizona	831	16-apr-2014	96.9
greenwich is a park in the city london	831	16-apr-2014	100.0

instances for many different relations

degree of certainty

NELL: http://rtw.ml.cmu.edu/rtw/

Graphs & Randomness

ProbLog, Phenetic, Prism, ICL, Probabilistic Databases, ...

- all based on a "random graph" model

Stochastic Logic Programs, ProPPR, PCFGs, ...

- based on a "random walk" model
- connected to PageRank
- not the subject of this talk !

Probabilistic Logic Programming

Distribution Semantics [Sato, ICLP 95]:
probabilistic choices + logic program
→ distribution over possible worlds

e.g., PRISM, ICL, ProbLog, LPADs, CP-logic, ...

multi-valued switches

probabilistic alternatives

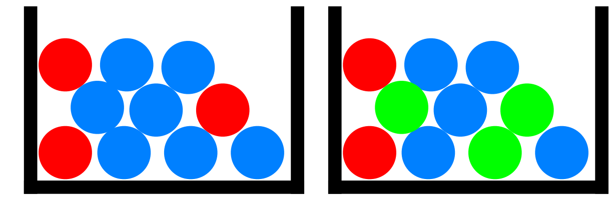
probabilistic facts

annotated disjunctions

causal-probabilistic laws

ProbLog by example:

A bit of gambling



- toss (biased) coin & draw ball from each urn
- win if (heads and a red ball) or (two balls of same color)

```
0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green);
                   0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).
```

probabilistic fact: heads is true with probability 0.4 (and false with 0.6)

annotated disjunction: first ball is red with probability 0.3 and blue with 0.7

probabilistic choices

annotated disjunction: second ball is red with probability 0.2, green with 0.3, and blue with 0.5

logical rule encoding background knowledge

consequences

Questions

```
0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green); 0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).
```

marginal probability

- Probability of `win`?

conditional probability

- Probability of `win` given `col(2,green)`?

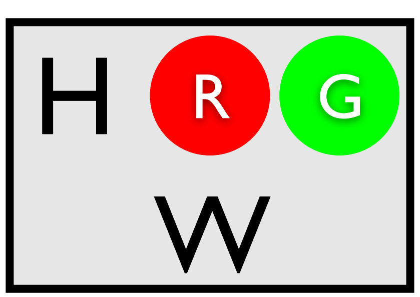
- Most probable world where `win` is true?

MPE inference

Possible Worlds

```
0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green); 0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).
```

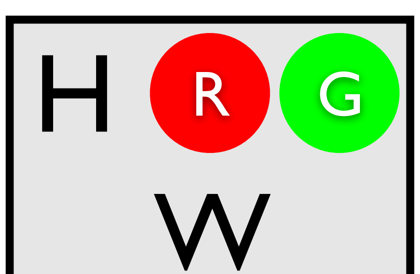
$0.4 \times 0.3 \times 0.3$



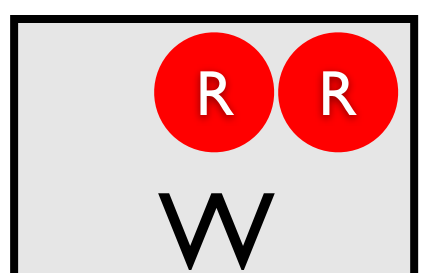
Possible Worlds

```
0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green); 0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).
```

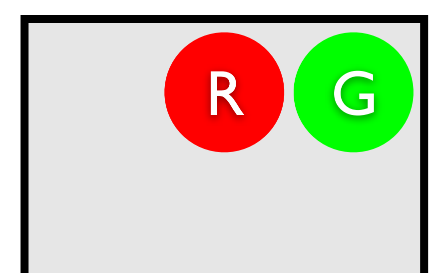
$0.4 \times 0.3 \times 0.3$



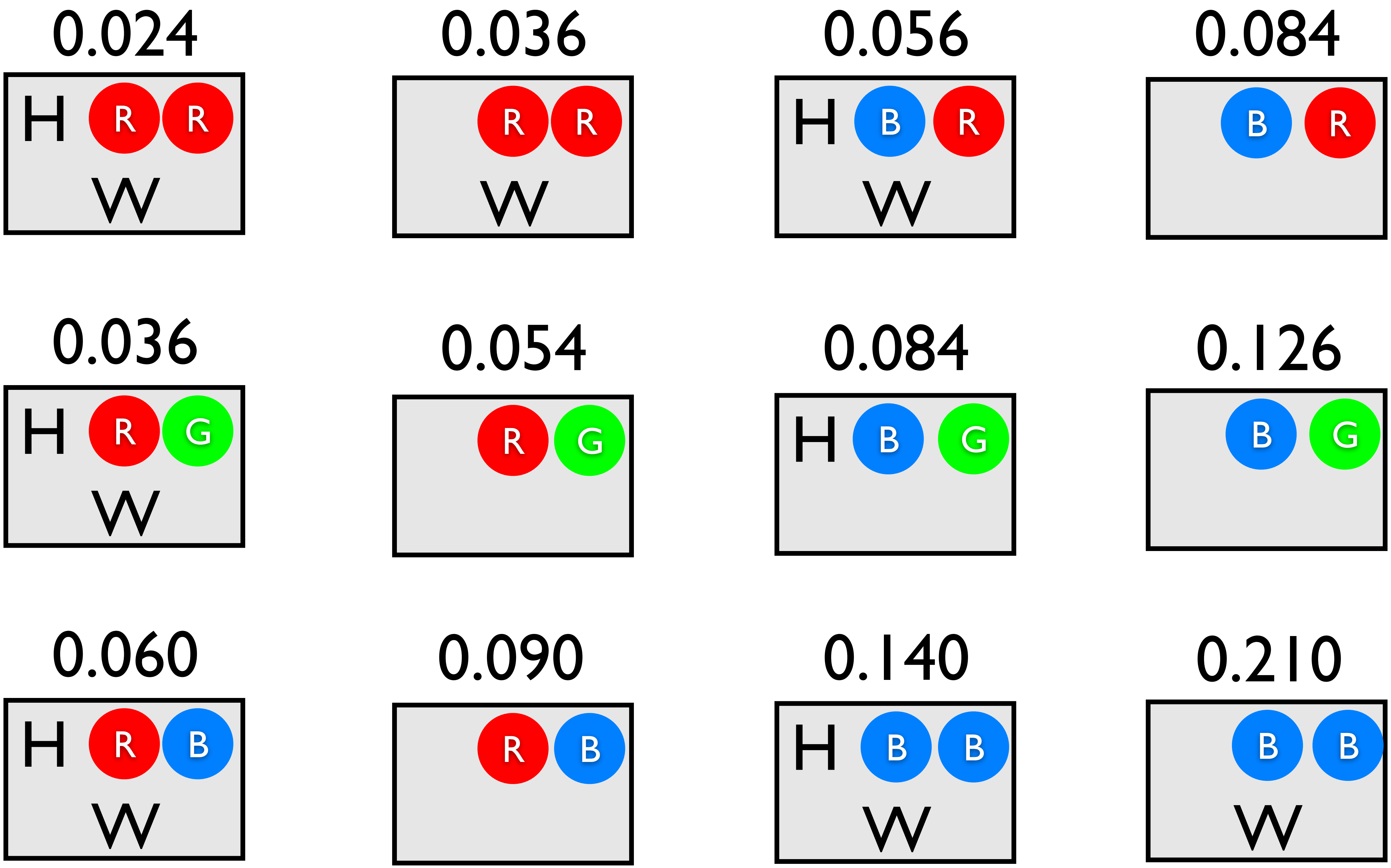
$(1-0.4) \times 0.3 \times 0.2$



$(1-0.4) \times 0.3 \times 0.3$

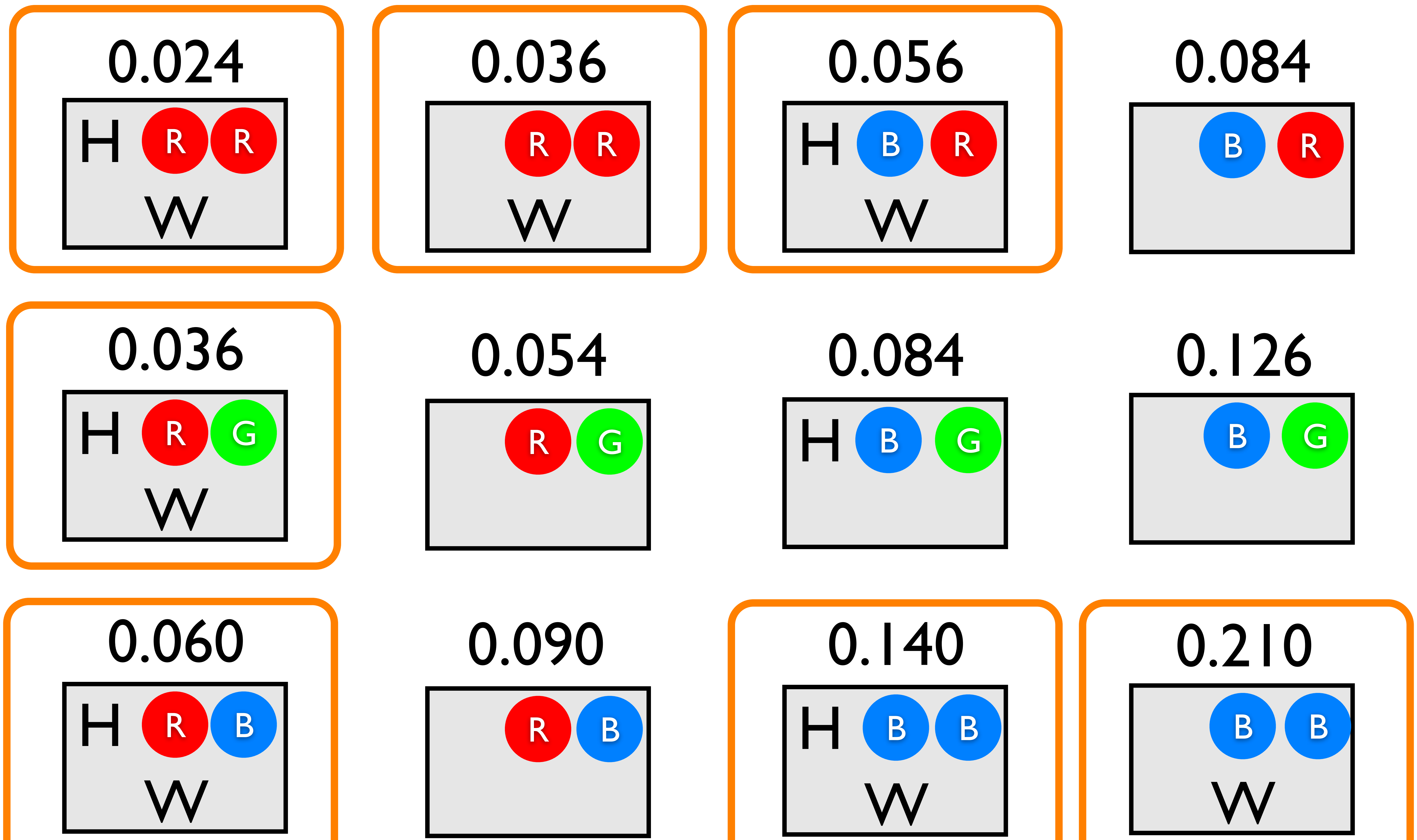


All Possible Worlds



$P(\underline{\text{win}}) = ?\sum = 0.562$

Marginal Probability

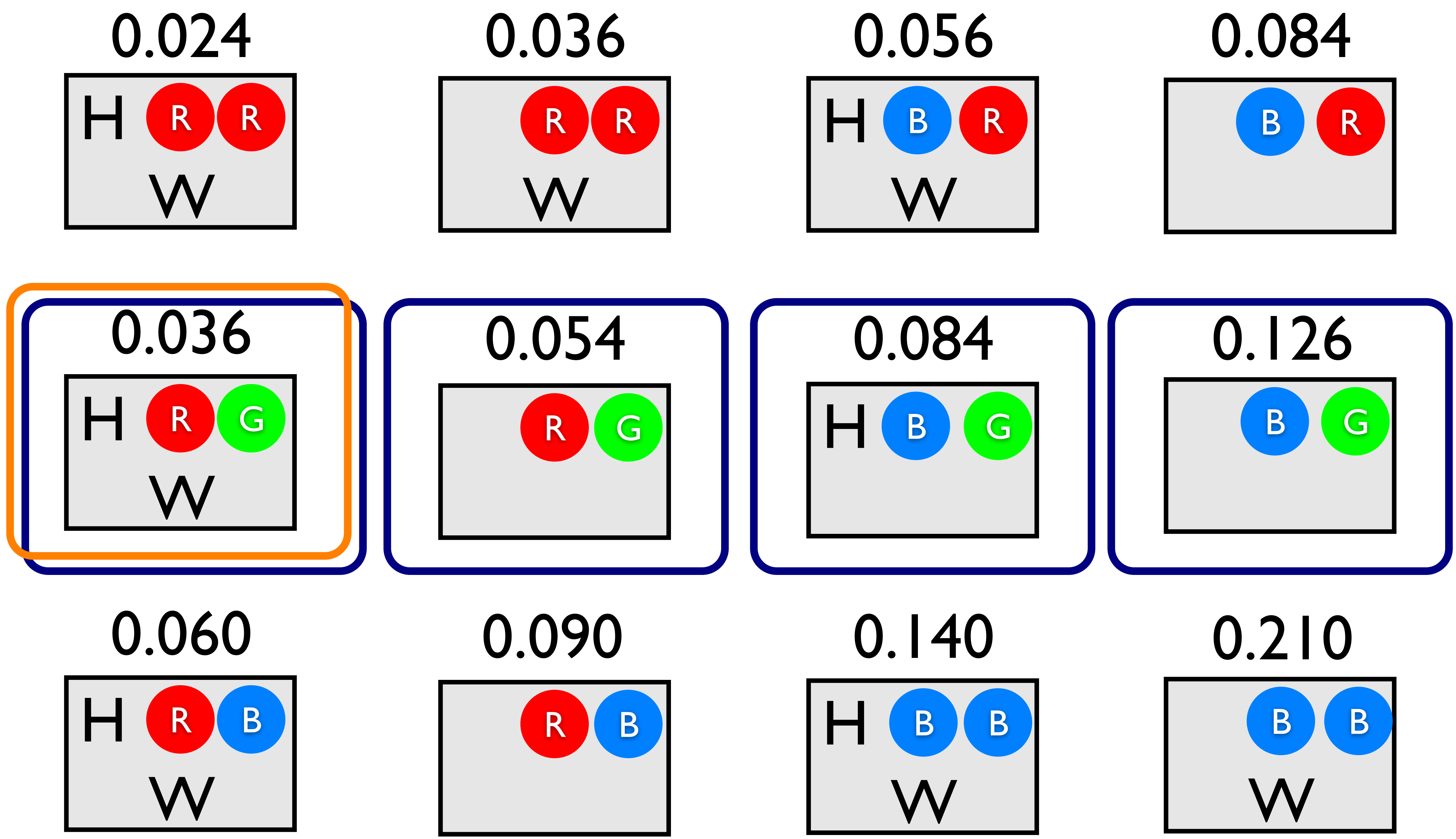


$P(win|col(2,green)) = ?\ \Sigma / \Sigma$

$= P(win \wedge col(2,green)) / P(col(2,green))$

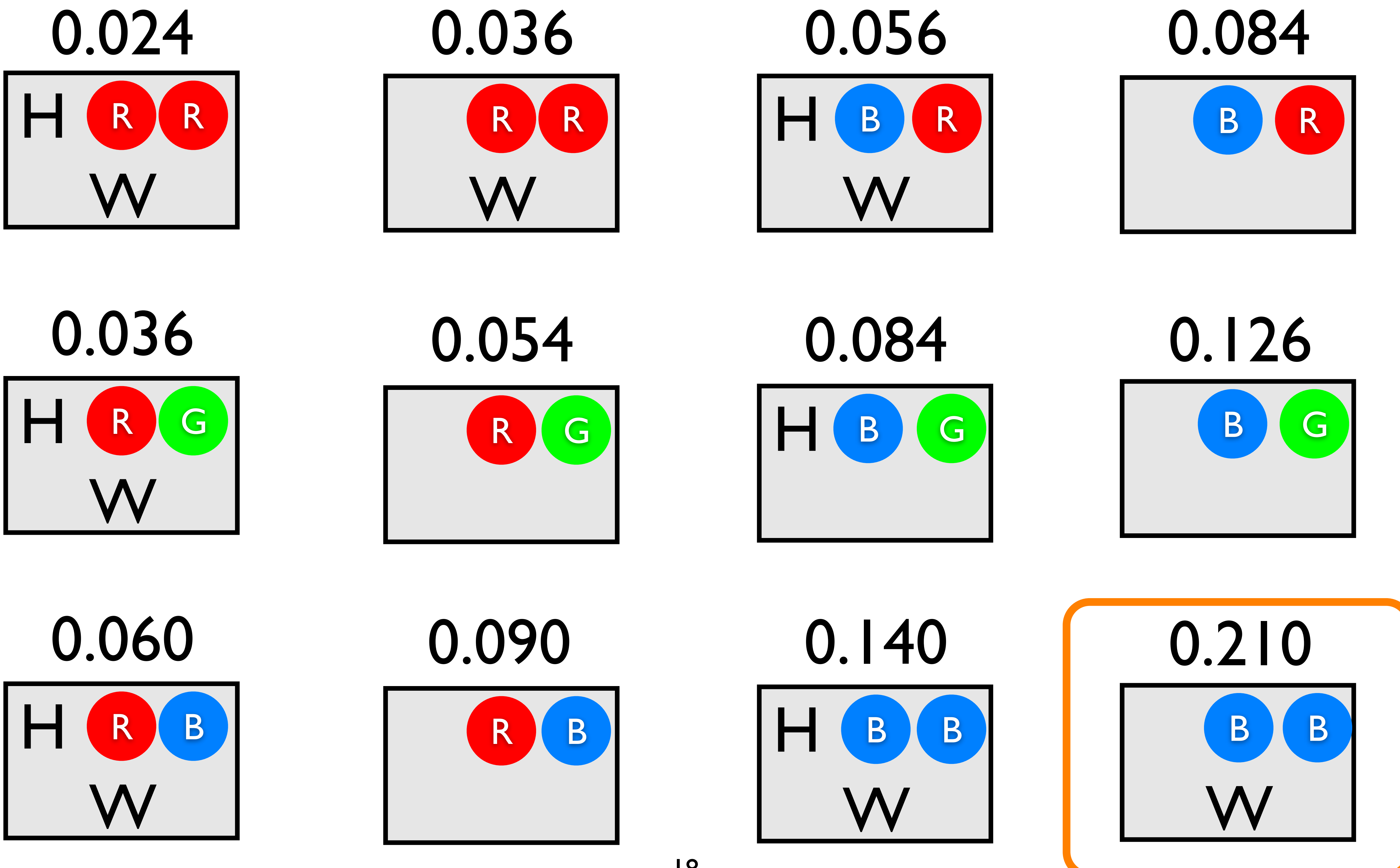
$= 0.036/0.3 = 0.12$

Conditional Probability



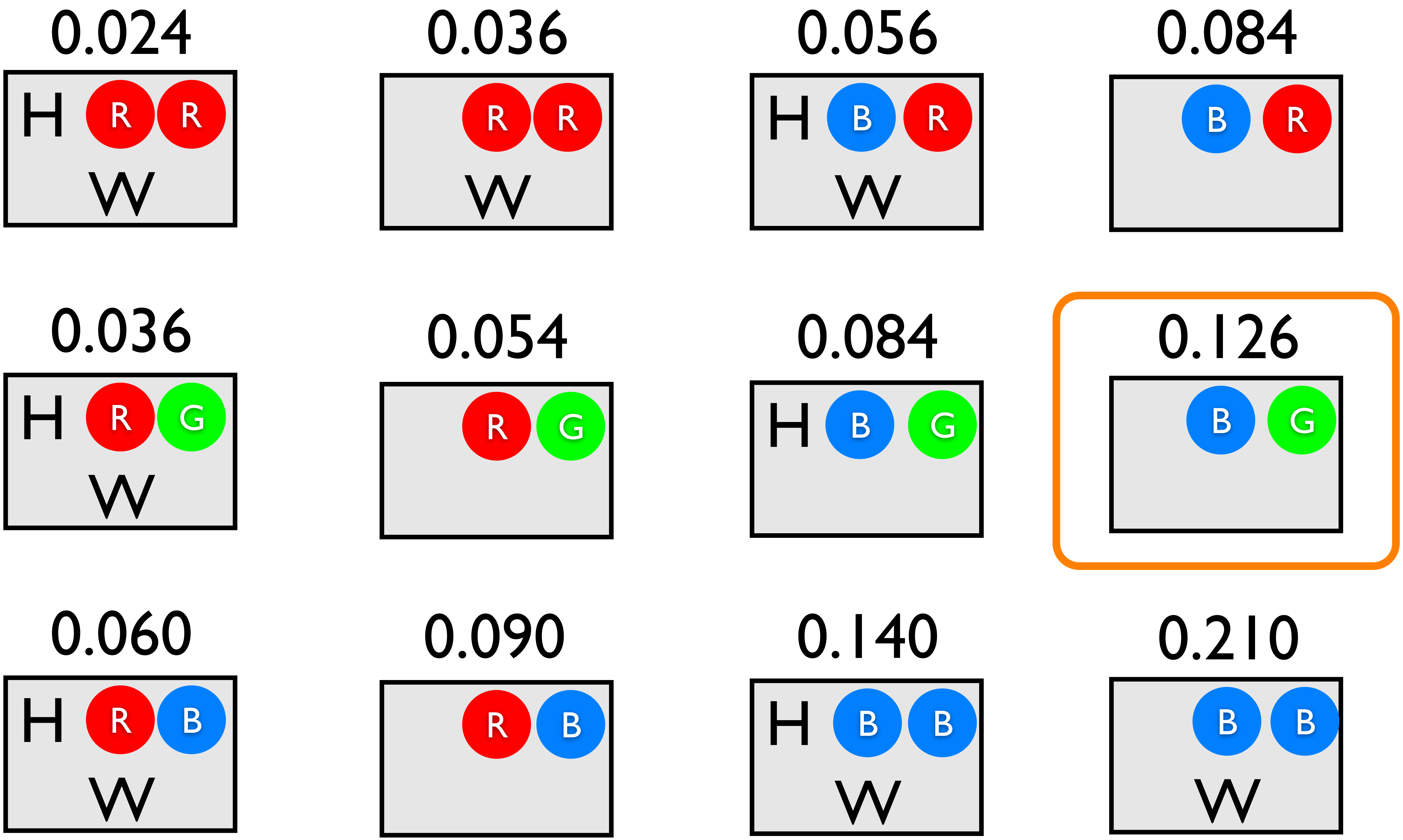
Most likely world where **`win`** is true?

MPE Inference



Most likely world where `col(2,blue)` is false?

MPE Inference



Distribution Semantics
(with probabilistic facts)

[Sato, ICLP 95]

query

sum over possible worlds
where Q is true

$$P(Q) = \sum_{F \cup R \models Q} \underbrace{\prod_{f \in F} p(f) \prod_{f \notin F} 1 - p(f)}_{\text{probability of possible world}}$$

subset of
probabilistic
facts

Prolog
rules

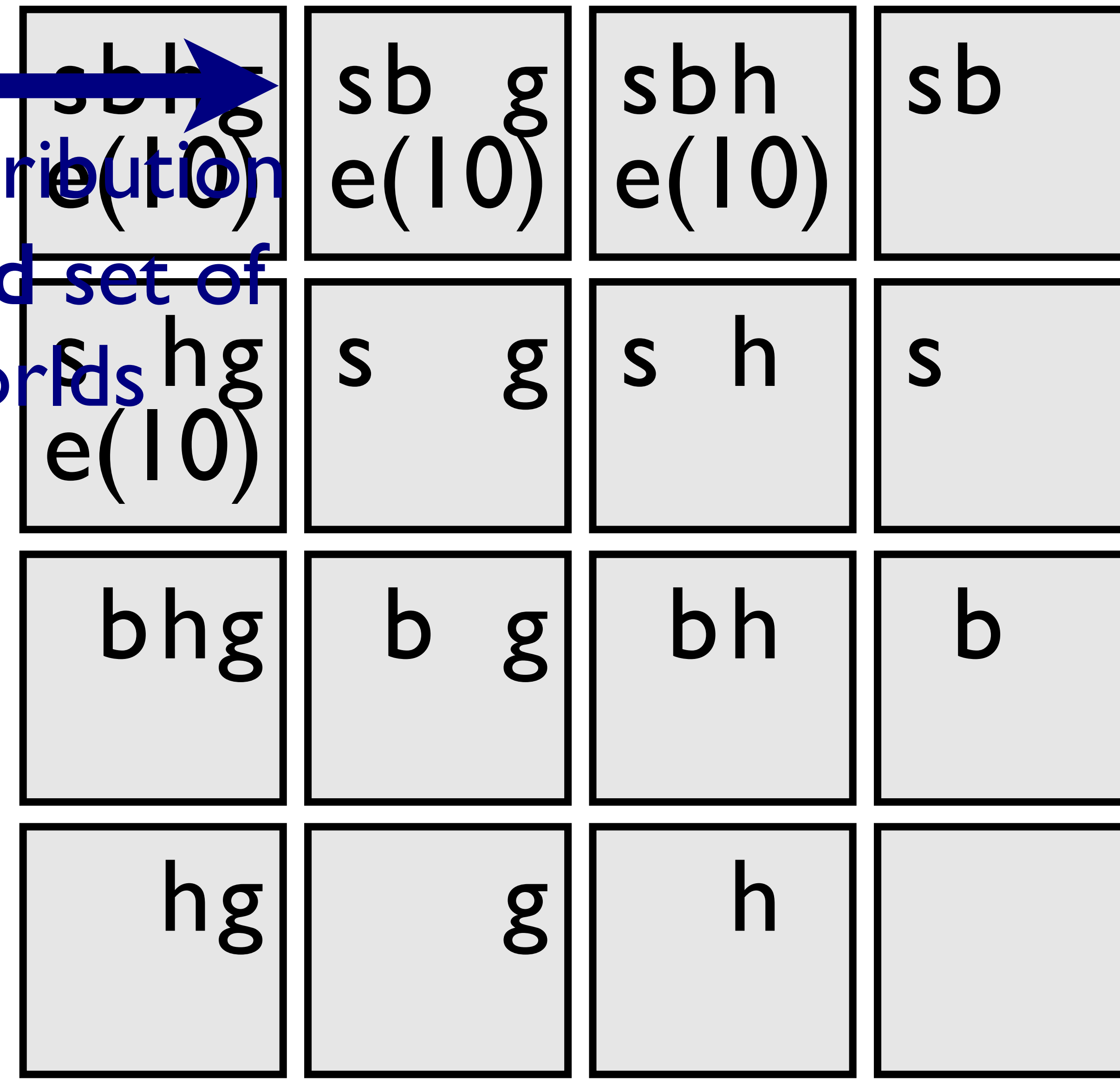
probability of
possible world

cProbLog: constraints on possible worlds

```
weight(skis,6).
weight(boots,4).
weight(helmet,3).
weight(gloves,2).

P::pack(Item) :-
  weight(Item,Weight),
  P is 1.0/Weight.

excess(Limit) :- ...

not excess(10).
pack(helmet) v pack(boots).
```

distribution over all possible worlds

normalized distribution over restricted set of possible worlds



constraints

as FOL formulas

treat as evidence

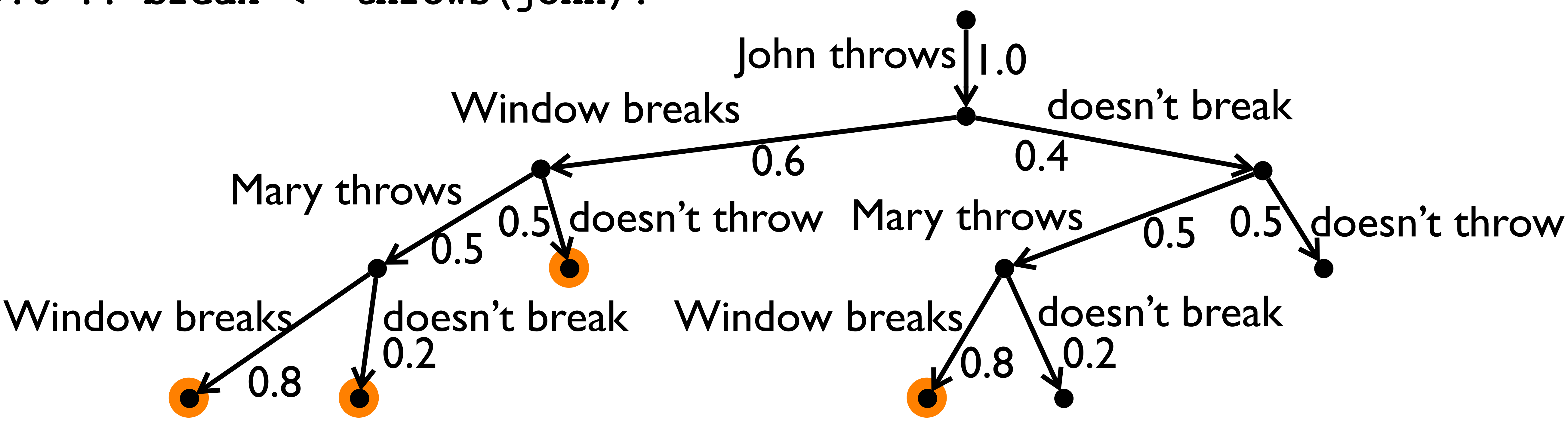
[Fierens et al, PP 12; Shterionov et al]

[Vennekens et al, ICLP 04]

Alternative view: CP-Logic

```
throws(john).
0.5::throws(mary).

0.8 :: break <- throws(mary).
0.6 :: break <- throws(john).
```

probabilistic causal laws



$$P(\text{break}) = 0.6\times0.5\times0.8 + 0.6\times0.5\times0.2 + 0.6\times0.5 + 0.4\times0.5\times0.8$$

CP-logic [Vennekens et al.]

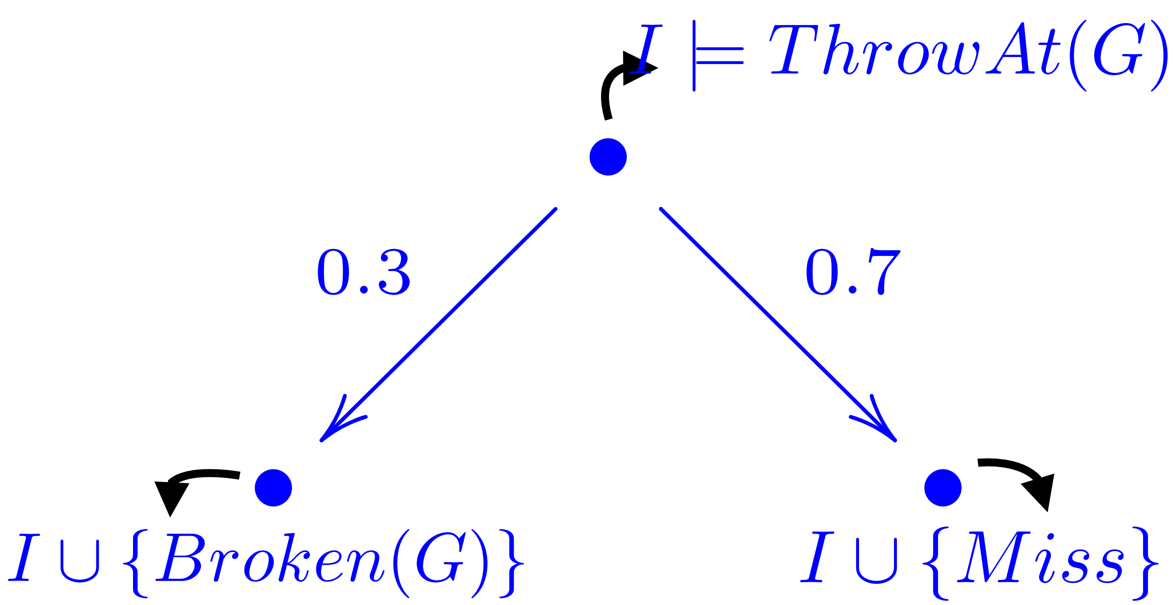
E.g., "throwing a rock at a glass breaks it with probability 0.3 and misses it with probability 0.7"

$$(Broken(G):0.3) \vee (Miss\ 0.7) \leftarrow ThrowAt(G).$$

ote that the actual non-deterministic event ("rock flying at glass") is implicit

Slides CP-logic courtesy Joost Vennekens

Semantics



(Broken(G) 0.3) ∨ (Miss 0.7)
← ThrowAt(G)

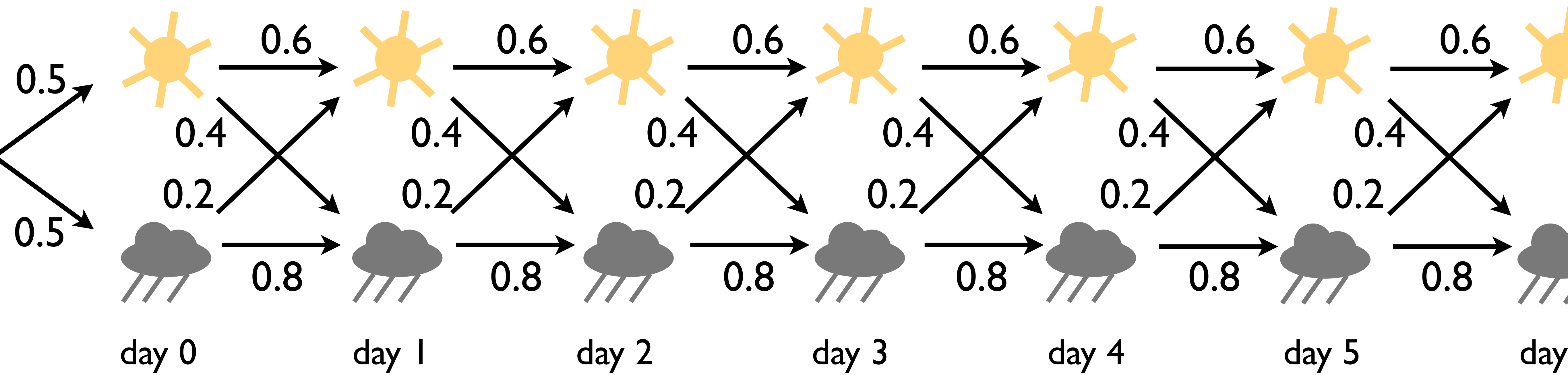
Probability tree is an execution model of theory iff:

- Each tree-transition matches causal law
- The tree cannot be extended
- Each execution model defines the <u>same</u> probability distribution over final states

Slides CP-logic courtesy Joost Vennekens

ProbLog by example:

Rain or sun?



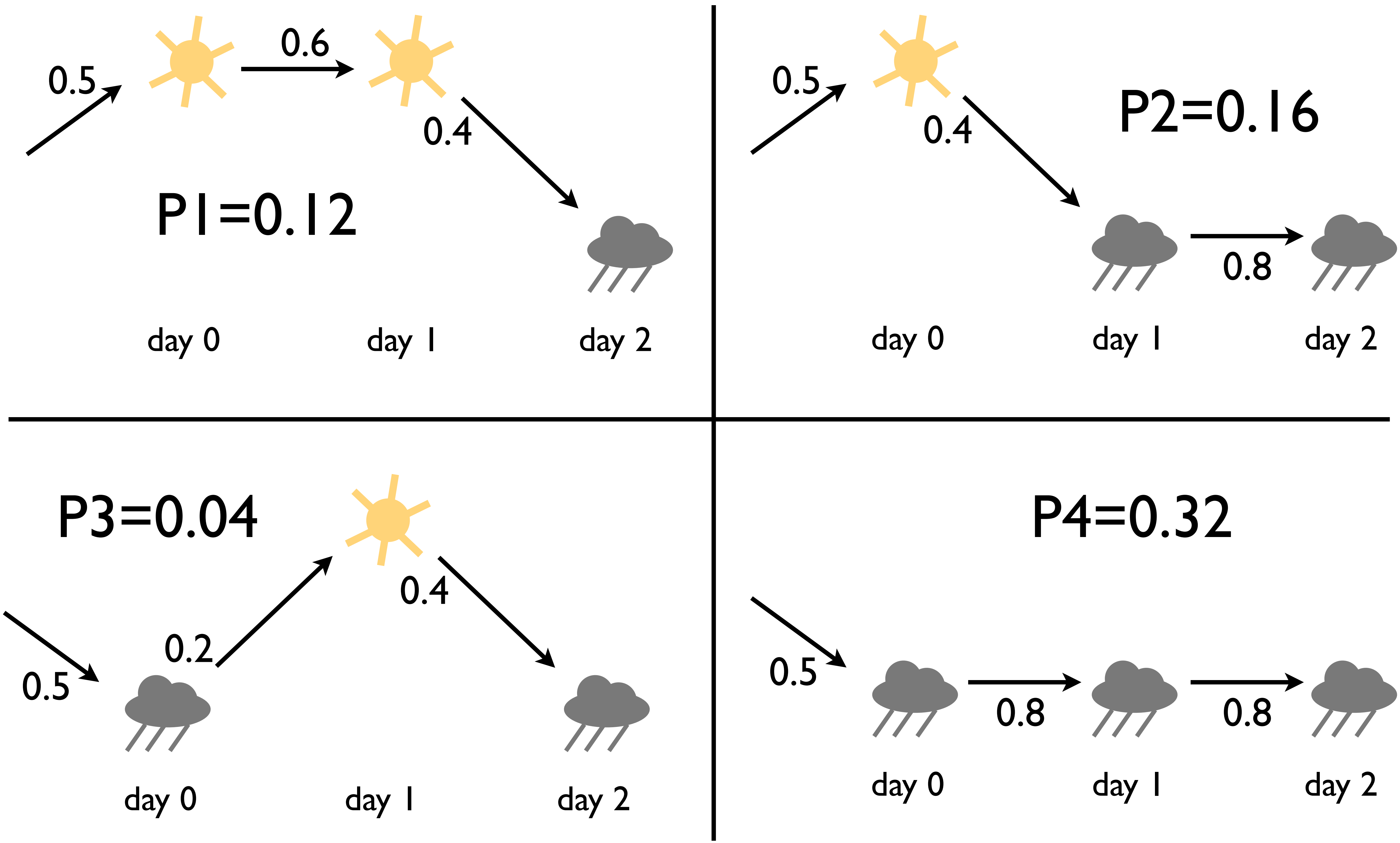

```
0.5::weather(sun,0) ; 0.5::weather(rain,0).

0.6::weather(sun,T) ; 0.4::weather(rain,T)
                        :- T>0, Tprev is T-1, weather(sun,Tprev).
0.2::weather(sun,T) ; 0.8::weather(rain,T)
                        :- T>0, Tprev is T-1, weather(rain,Tprev).
```

infinite possible worlds! BUT: finitely many partial worlds suffice to answer any given ground query

Possible worlds

```
?- weather(rain,2).
```

P=P1+P2+P3+P4



Distributional Clauses (DC)

Defines a generative process (as for CP-logic)
Tree can become infinitely wide
Sampling
Well-defined under reasonable

- Discrete- and continuou … om variables

random variable with Gaussian distribution

```
length(Obj) ~ gaussian(6.0,0.45) :- type(Obj,glass).
stackable(OBot,OTop) :-
      ≃length(OBot) ≥ ≃length(OTop),
      ≃width(OBot) ≥ ≃width(OTop).
ontype(Obj,plate) ~ finite([0 : glass, 0.0024 : cup,
                            0 : pitcher, 0.8676 : plate,
                            0.0284 : bowl, 0 : serving,
                            0.1016 : none])
                   :- obj(Obj), on(Obj,O2), type(O2,plate).
```

comparing values of random variables

random variable with discrete distribution

[Gutmann et al, TPLP 11; Nitti et al, IROS 13]

Distributional Clauses (DC)

- Defines a generative process (as for CP-logic)
- Tree can become infinitely wide
  - Sampling …
- Well-defined under reasonable assumptions

ProbLog

- **probabilistic choices** + their **consequences**
- probability distribution over **possible worlds**
- how to efficiently answer **questions**?
  - most probable world (MPE inference)
  - probability of query (computing marginals)
  - probability of query given evidence

Summary: ProbLog Syntax

- input database: ground facts

```
person(bob).
```

- probabilistic facts

```
0.5::stress(bob).
```

- annotated disjunctions

```
0.5::stress(X) :- person(X).
0.4::a(X); 0.3::b(X); 0.2::c(X); 0.1::d(X) :- q(X).
0.5::weather(sun,0) ; 0.5::weather(rain,0).
```

- flexible probabilities

```
P::pack(Item) :- weight(Item,W), P is 1.0/W.
```

- Prolog clauses

```
smokes(X) :- influences(Y,X), smokes(Y).
excess([I|R],Limit) :- \+pack(I), excess(R,Limit).
```

Probabilistic Logic Programming

Distribution Semantics [Sato, ICLP 95]:
probabilistic choices + logic program
→ distribution over possible worlds

OVERVIEW paper [Kimmig, De Raedt, Arxiv, MLJ 15]

e.g., PRISM, ICL, ProbLog, LPADs, CP-logic, ...

multi-valued switches

probabilistic alternatives

probabilistic facts

annotated disjunctions

causal-probabilistic laws

Probabilistic databases

programming versus database query language
different types of queries

ProducesProduct

Company	Product	P
sony	walkman	0.96
microsoft	mac_os_x	0.96
ibm	personal_computer	0.96
microsoft	mac_os	0.9
adobe	adobe_indesign	0.9
adobe	adobe_dreamweaver	0.87
...	...	...

HeadquarteredIn

Company	City	P
microsoft	redmond	1.00
ibm	san_jose	0.99
emirates_airlines	dubai	0.93
honda	torrance	0.93
horizon	seattle	0.93
egyptair	cairo	0.93
adobe	san_jose	0.93
...	...	...

```
select x.Product, x.Company
from ProducesProduct x, HeadquarteredIn y
where x.Company=y.Company and
y.City='san_jose'
```

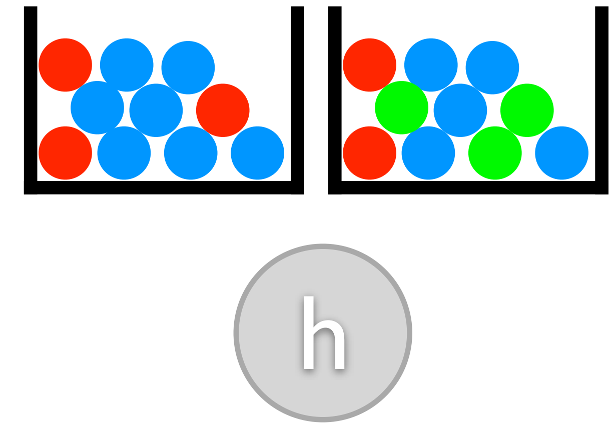

[Example from Suciu et al 2011]

Probabilistic Programs

- Distributional clauses / PLP similar in spirit
  - to e.g. BLOG, ... but embedded in existing logic and programming language
  - to e.g. Church but use of logic instead of functional programming ...
    - natural possible world semantics and link with prob. databases.
    - somewhat harder to do meta-programming

Church by example:

A bit of gambling



- toss (biased) coin & draw ball from each urn
- win if (heads and a red ball) or (two balls of same color)

```
(define heads (mem (lambda () (flip 0.4))))

(define color1 (mem (lambda () (if (flip 0.3) 'red 'blue))))

(define color2 (mem (lambda ()
                 (multinomial '(red green blue) '(0.2 0.3 0.5)))))

(define redball (or (equal? (color1) 'red) (equal? (color2) 'red)))

(define win1 (and (heads) redball))

(define win2 (equal? (color1) (color2)))

(define win (or win1 win2))
```

Markov Logic

Key differences

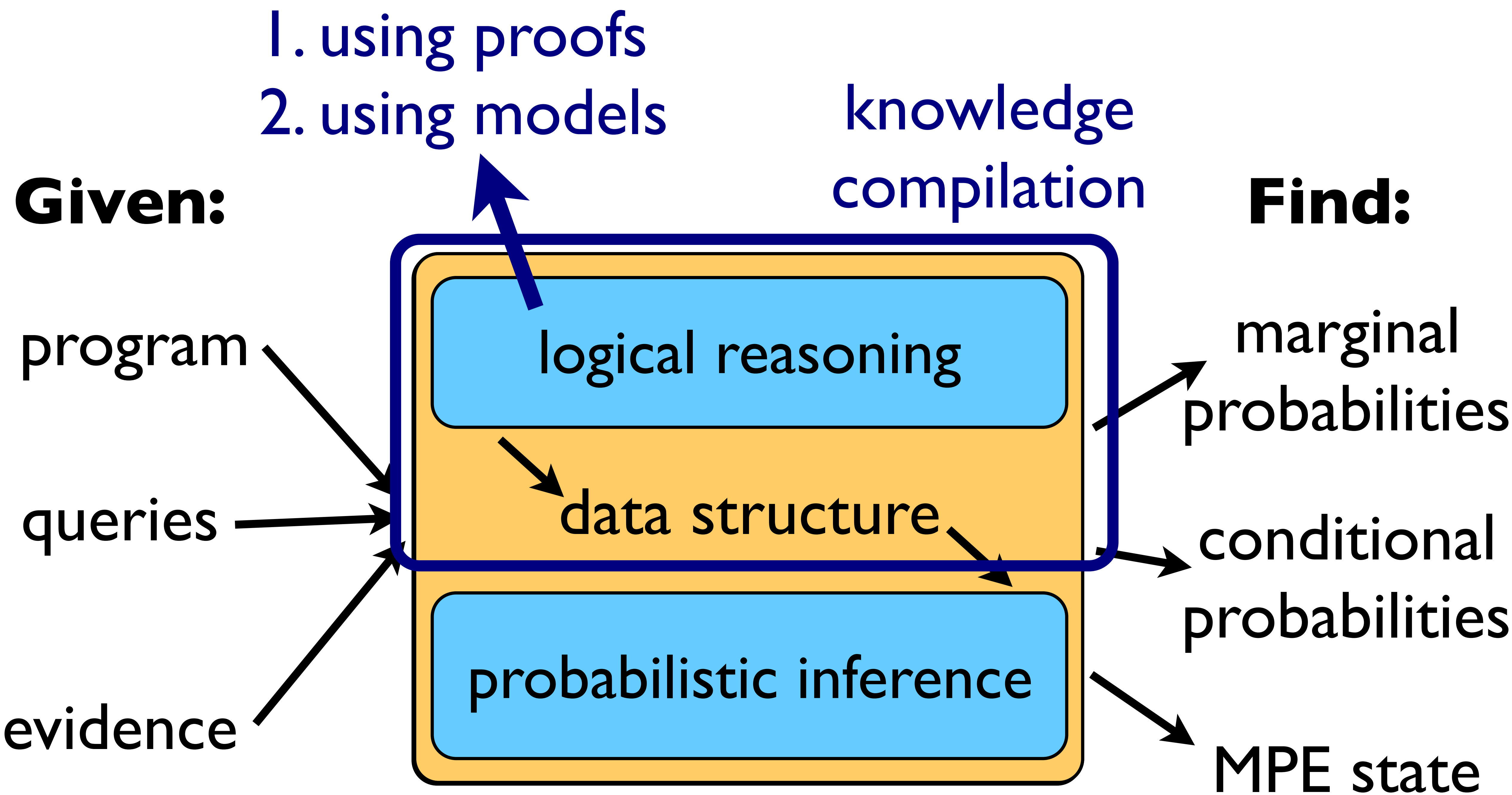
- programming language
- Pro(b)log uses least-fix point semantics
  - can express transitive closure of relation
  - this cannot be expressed in FOL (and Markov Logic), requires second order logic
  - p(X,Y) :- p(X,Z), p(Z,Y).

PART II: Inference

Inference in PLP

- As in Prolog and logic programming
  - proof-based
- As in Answer Set Programming
  - model based
- As in Probabilistic Programming
  - sampling

Inference


1. using proofs
2. using models
knowledge
compilation
Given:
Find:
program
queries
evidence
logical reasoning
data structure
probabilistic inference
marginal
probabilities
conditional
probabilities
MPE state

Proofs in ProbLog

```
0.8::stress(ann).
0.6::influences(ann,bob).
0.2::influences(bob,carl).

smokes(X) :- stress(X).
smokes(X) :-
      influences(Y,X),
      smokes(Y).
```

```
?- smokes(carl).
?- stress(carl).
?- influences(Y,carl),smokes(Y).
```

Y=bob

```
?- smokes(bob).
?- stress(bob).
?- influences(Y1,bob),smokes(Y1).
```

Y1=ann

```
?- smokes(ann).
?- stress(ann).
?- influences(Y2,ann),smokes(Y2).
□
```

`influences(bob,carl)&influences(ann,bob)&stress(ann)`

probability of proof = $0.2 \times 0.6 \times 0.8 = 0.096$

Proofs in ProbLog

```
0.8::stress(ann).
0.4::stress(bob).
0.6::influences(ann,bob).
0.2::influences(bob,carl).

smokes(X) :- stress(X).
smokes(X) :-
        influences(Y,X),
        smokes(Y).
```

`?- smokes(carl).`

`?- stress(carl).`

Y=bob

`?- influences(Y,carl),smokes(Y).`

`?- smokes(bob).`

Y1=ann

`influences(bob,carl)`
`& stress(bob)`

`?- stress(bob).`

`?- influences(Y1,bob),smokes(Y1).`

0.2×0.4
$= 0.08$

□

`?- smokes(ann).`

`?- stress(ann).`

`?- influences(Y2,ann),smokes(Y2).`

□

`influences(bob,carl)`
`& influences(ann,bob)`
`& stress(ann)`

$0.2 \times 0.6 \times 0.8$
$= 0.096$

proofs overlap!
cannot sum probabilities
(disjoint-sum-problem)

Disj

solution: knowledge compilation

possible worlds

influences(bob,carl) &
influences(ann,bob) & stress(ann)

```
infl(bob,carl) &   infl(ann,bob) &   st(ann) & \+st(bob)   0.0576
infl(bob,carl) &   infl(ann,bob) &   st(ann) &   st(bob)   0.0384
infl(bob,carl) & \+infl(ann,bob) &   st(ann) &   st(bob)   0.0256
infl(bob,carl) &   infl(ann,bob) & \+st(ann) &   st(bob)   0.0096
infl(bob,carl) & \+infl(ann,bob) & \+st(ann) &   st(bob)   0.0064
...
```

influences(bob,carl) & stress(bob)

$\sum = 0.1376$

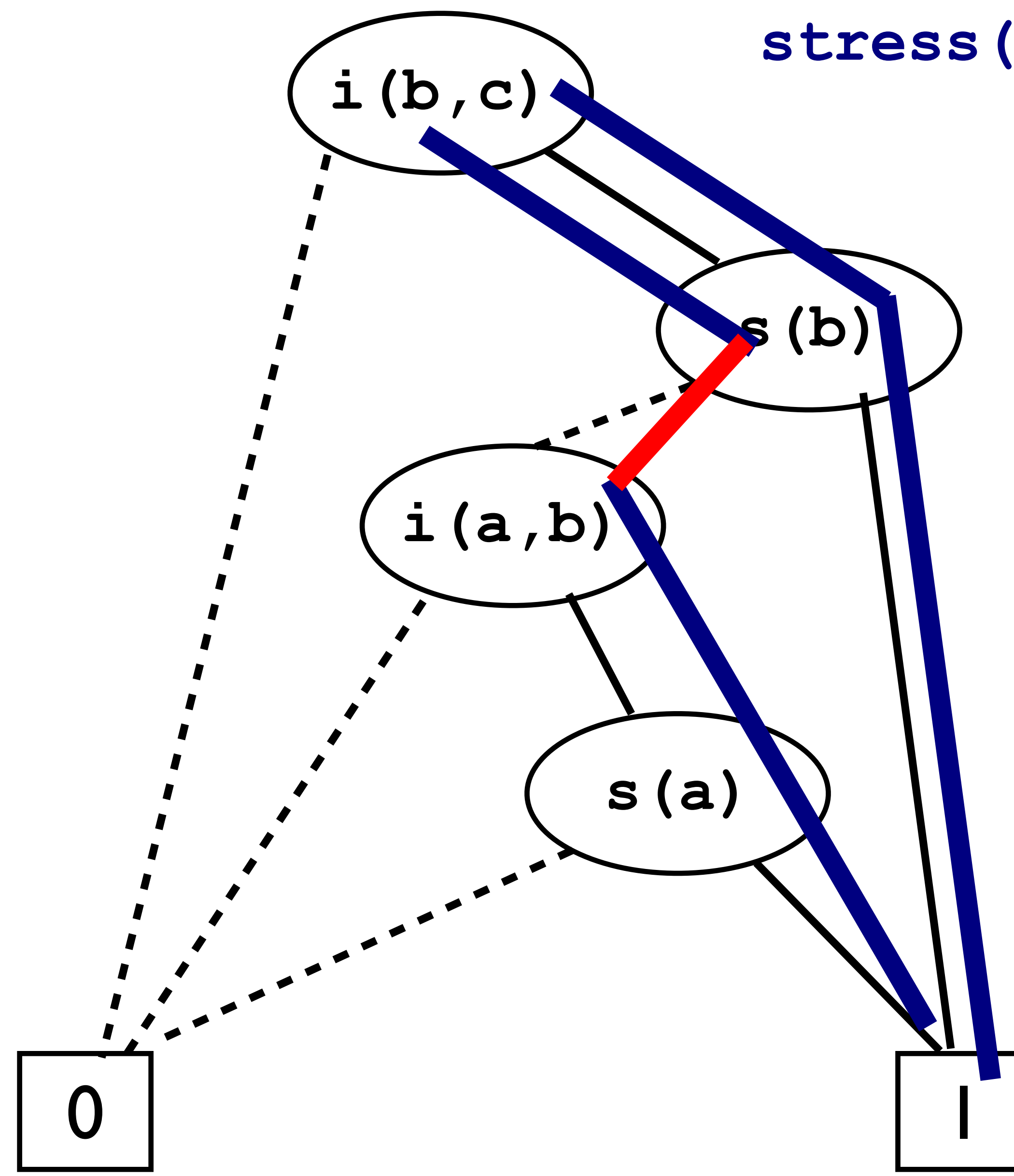
sum of proof probabilities: 0.096+0.08 = 0.1760

Binary Decision Diagrams

[Bryant 86]

- compact graphical representation of Boolean formula
- automatically disjoins proofs
- popular in many branches of CS

`influences(bob,carl) & stress(bob)`



`influences(bob,carl) & influences(ann,bob) & stress(ann) & not stress(bob)`

Current Approach
(ProbLog2)

```
0.4::heads(1).
0.7::heads(2).
0.5::heads(3).
win :- heads(1).
win :- heads(2),
            heads(3).
```

Find relevant ground program for queries & evidence

`win`

Weighted CNF

use weighted model counting / satisfiability

```
win :- heads(1).
win :- heads(2), heads(3).
```

$$\text{win} \leftrightarrow h(1) \vee (h(2) \wedge h(3))$$

may require loop-breaking

$$(\neg\text{win} \vee h(1) \vee h(2))$$
$$\wedge\ (\neg\text{win} \vee h(1) \vee h(3))$$
$$\wedge\ (\text{win} \vee \neg h(1))$$
$$\wedge\ (\text{win} \vee \neg h(2) \vee \neg h(3))$$

use standard tool

$h(1) \rightarrow 0.4$	$h(2) \rightarrow 0.7$	$h(3) \rightarrow 0.5$
$\neg h(1) \rightarrow 0.6$	$\neg h(2) \rightarrow 0.3$	$\neg h(3) \rightarrow 0.5$

[Fierens et al, TPLP 14]

ProbLog → CNF

```
0.8::stress(ann).
0.4::stress(bob).
0.6::influences(ann,bob).
0.2::influences(bob,carl).

smokes(X) :- stress(X).
smokes(X) :-
      influences(Y,X),
      smokes(Y).
```

```
?- smokes(carl).
```

- Find relevant ground rules by backward reasoning

```
smokes(carl) :- influences(bob,carl),smokes(bob).
smokes(bob) :- stress(bob).
smokes(bob) :- influences(ann,bob),smokes(ann).
smokes(ann) :- stress(ann).
```

- Convert to propositional logic formula

may require loop-breaking

$$\begin{aligned} & sm(c) \leftrightarrow (i(b,c) \wedge sm(b)) \\ \wedge\ & sm(b) \leftrightarrow (st(b) \vee (i(a,b) \wedge sm(a))) \\ \wedge\ & sm(a) \leftrightarrow st(a) \end{aligned}$$

- Rewrite in CNF (as usual)

Weighte

$$P(Q) = \sum_{F \cup R \models Q} \prod_{f \in F} p(f) \prod_{f \notin F} 1 - p(f)$$

propositional formula in conjunctive normal form (CNF)

given by ProbLog program & query

$$WMC(\phi) = \sum_{I_V \models \phi} \prod_{l \in I_V} w(l)$$

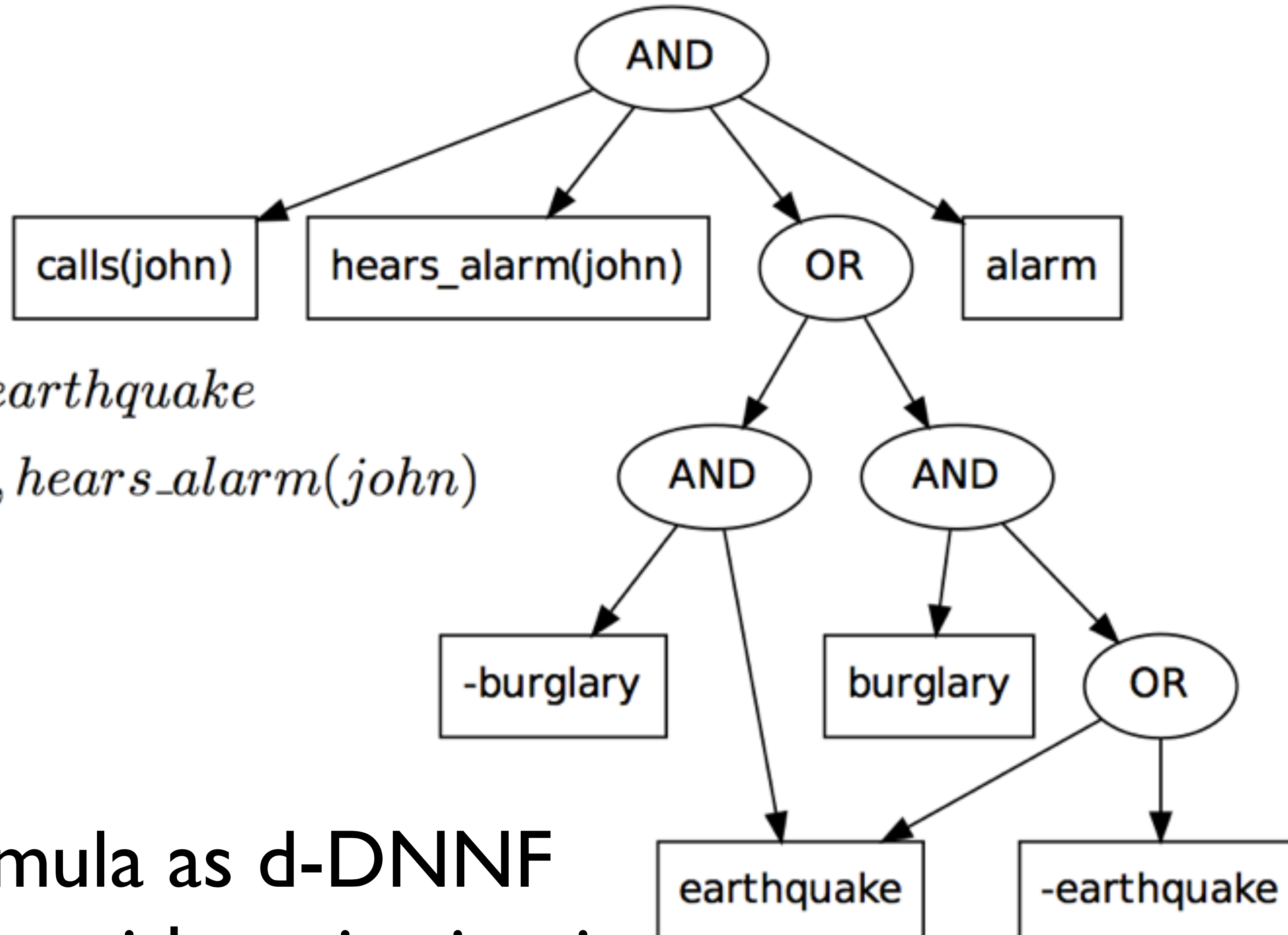
interpretations (truth value assignments) of propositional variables

possible worlds

weight of literal

for p::f,
w(f) = p
w(not f) = 1−p

WMC using d-DNNFs



$$alarm \leftrightarrow burglary \vee earthquake$$

$$calls(john) \leftrightarrow alarm, hears\_alarm(john)$$

$$calls(john)$$

1. represent formula as d-DNNF
2. transform into arithmetic circuit
3. evaluate bottom-up

[Figure: Fierens et al, TPLP 14]

Current Approach
(ProbLog2)

```
0.4::heads(1).
0.7::heads(2).
0.5::heads(3).
win :- heads(1).
win :- heads(2),
            heads(3).
```

Find relevant ground program for queries & evidence

`win`

Weighted CNF

use weighted model counting / satisfiability

```
win :- heads(1).
win :- heads(2), heads(3).
```

$$\text{win} \leftrightarrow h(1) \vee (h(2) \wedge h(3))$$

may require loop-breaking

$$(\neg\text{win} \vee h(1) \vee h(2)) \wedge (\neg\text{win} \vee h(1) \vee h(3)) \wedge (\text{win} \vee \neg h(1)) \wedge (\text{win} \vee \neg h(2) \vee \neg h(3))$$

use standard tool

$h(1) \rightarrow 0.4$ $\quad h(2) \rightarrow 0.7$ $\quad h(3) \rightarrow 0.5$

$\neg h(1) \rightarrow 0.6$ $\quad \neg h(2) \rightarrow 0.3$ $\quad \neg h(3) \rightarrow 0.5$

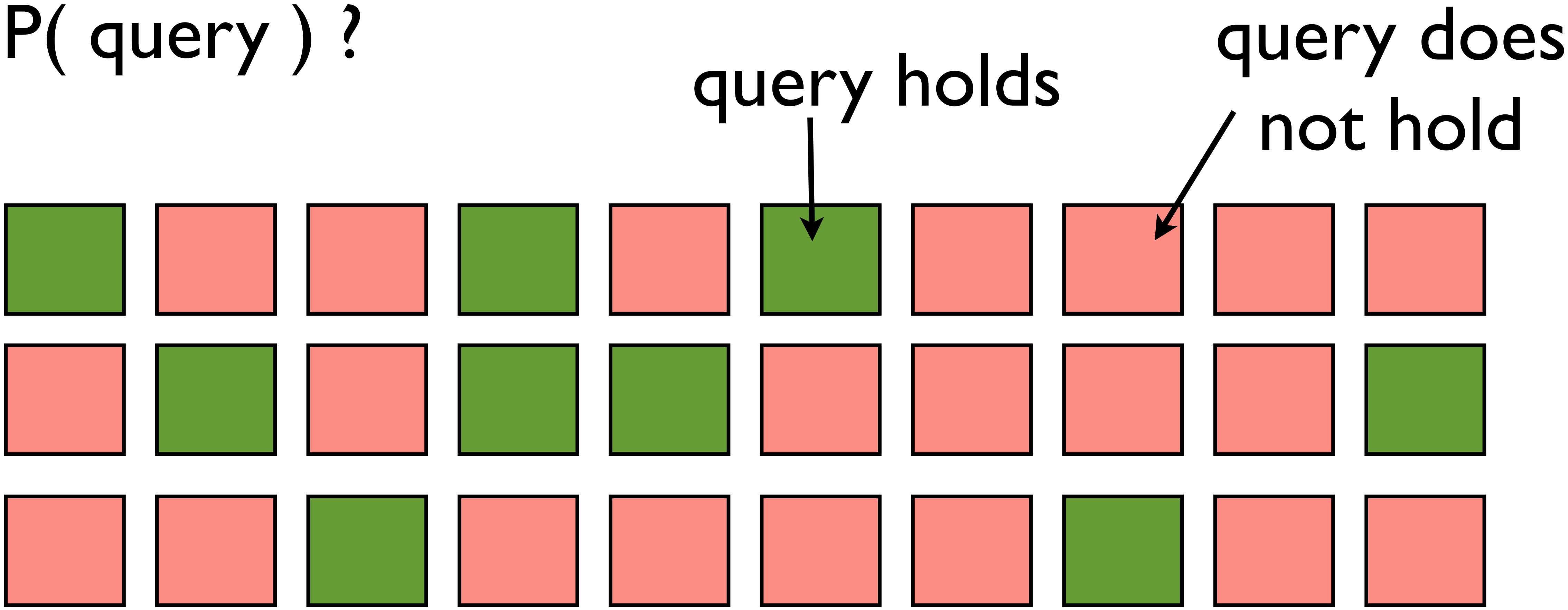
[Fierens et al, TPLP 14]

Inference for DC


```
n ~ uniform([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]).
color(X) ~ uniform([grey, blue, black]) ← material(X) ~= metal.
color(X) ~ uniform([black, brown]) ← material(X) ~=wood.
material(X)~finite([0.3:wood,0.7:metal])←n~=N,between(1,N,X).
drawn(Y) ~ uniform(L) ← n ~= N, findall(X, between(1, N, X), L).
size(X) ~ beta(2, 3) ← material(X) ~= metal.
size(X) ~ beta(4, 2) ← material(X) ~= wood.
```

Sampling

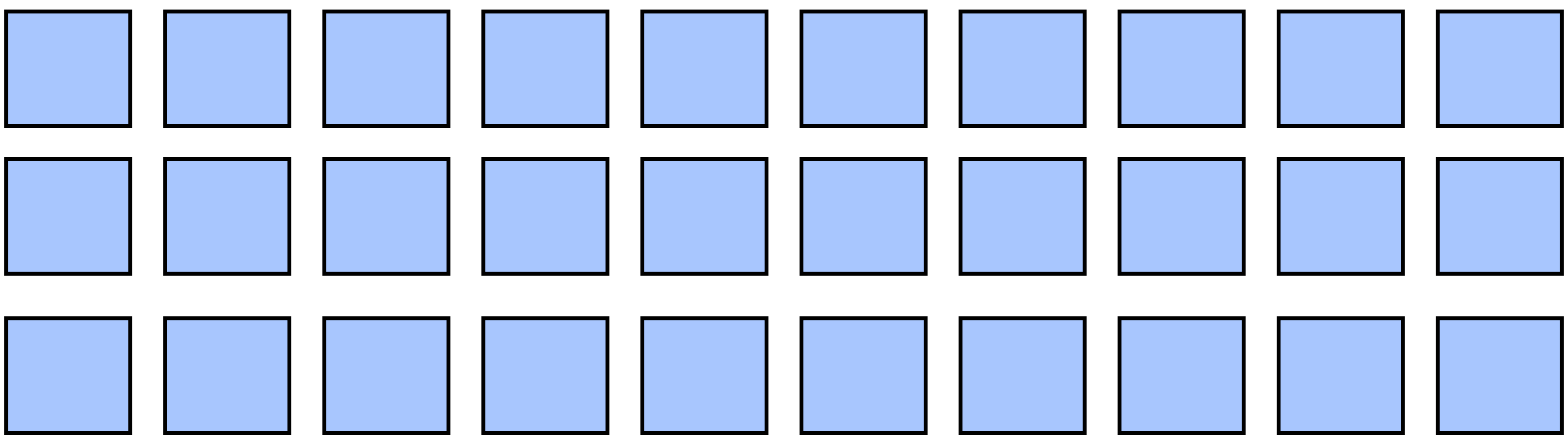
- P(query) ?



$$P(\text{query}) \approx \frac{\#\ \text{query holds}}{\#\ \text{worlds sampled}}$$

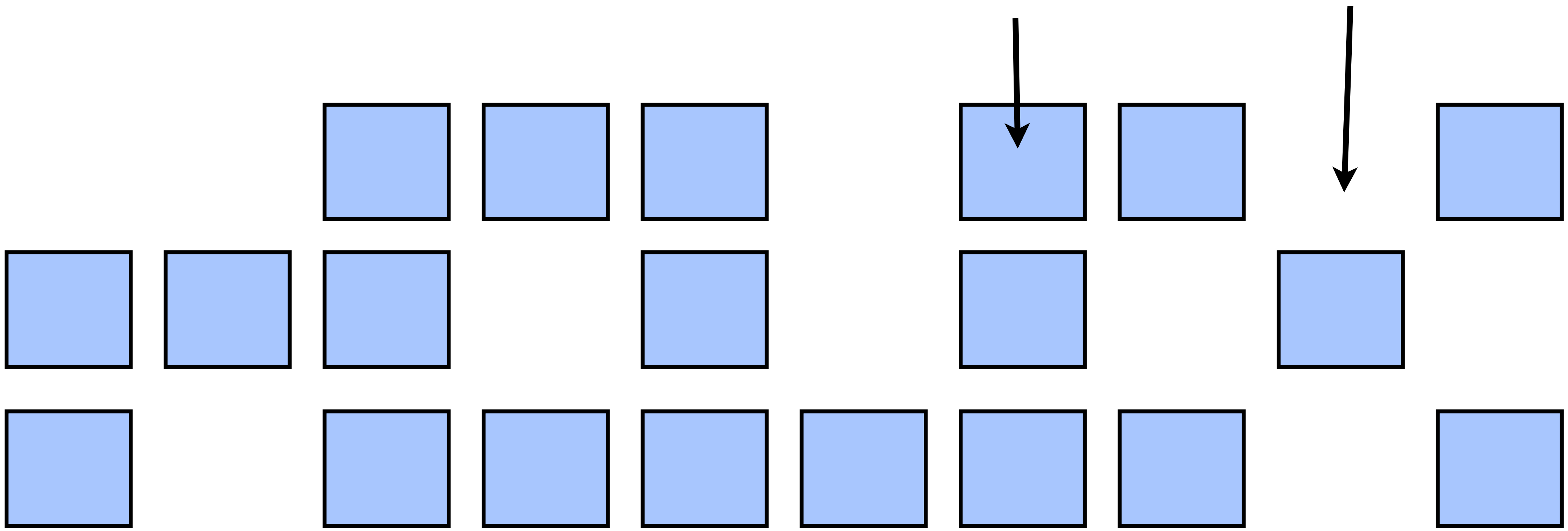
Rejection Sampling

- P(query | evidence) ?

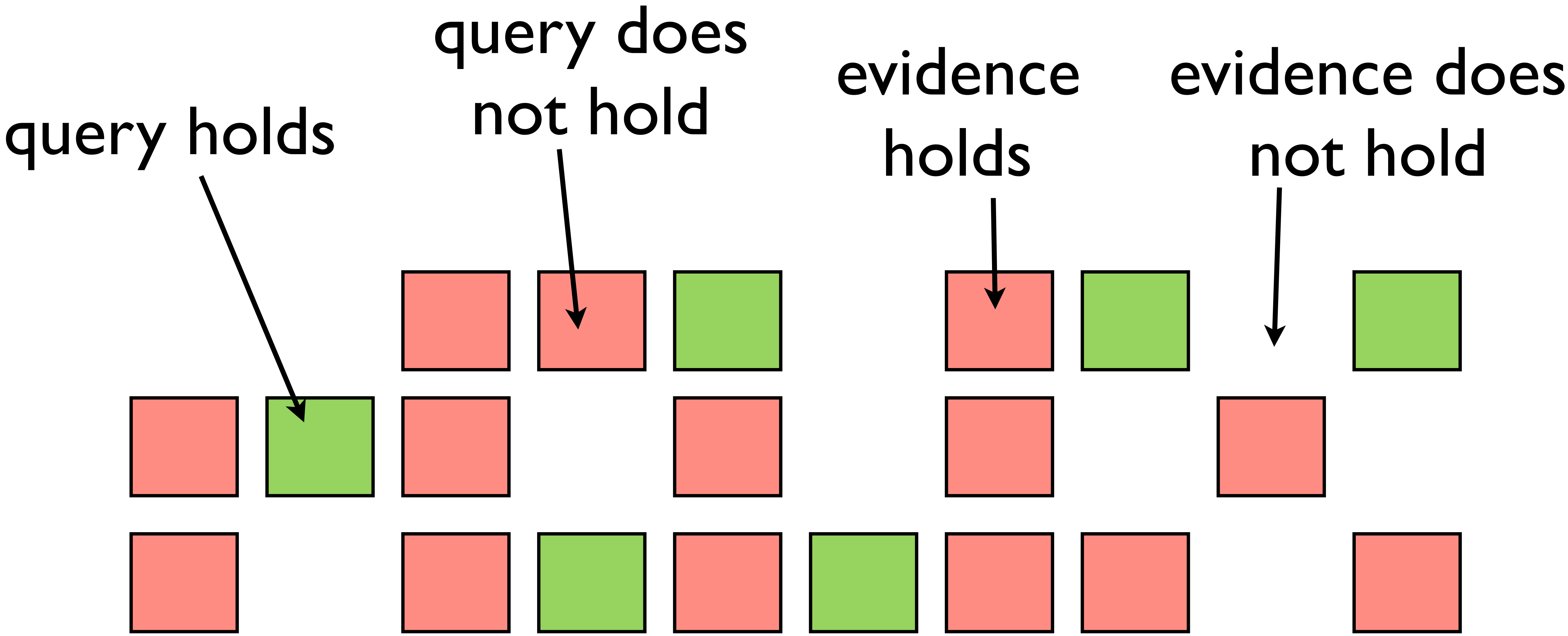


Rejection Sampling

- P(query | evidence) ?



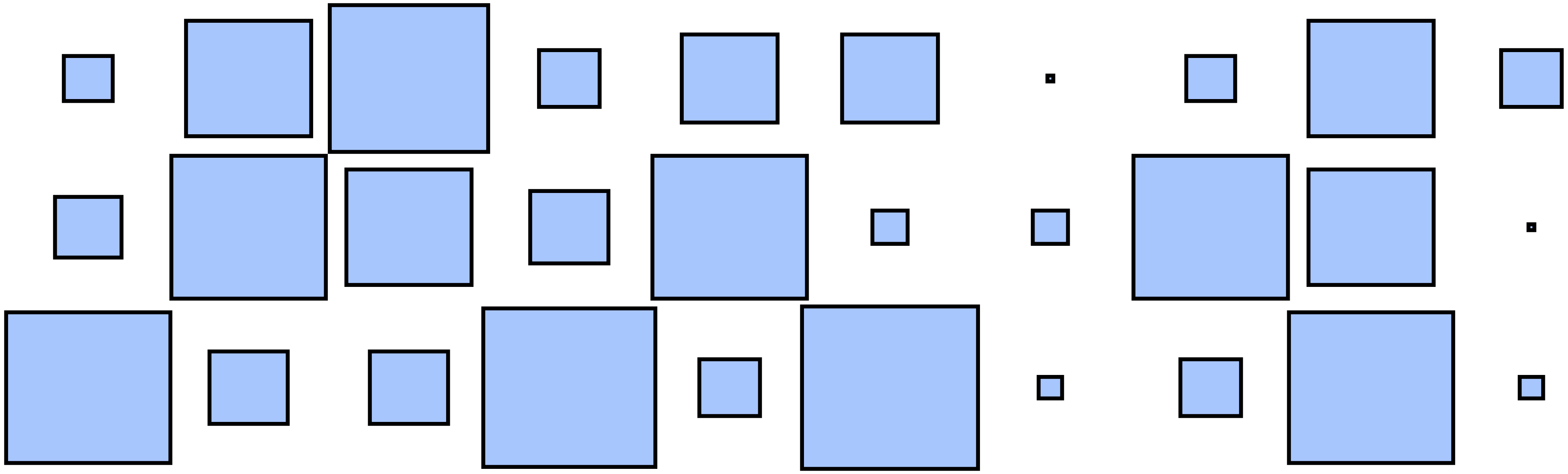
Rejection Sampling



$$P(\text{query} \mid \text{evidence}) \approx \frac{\#\ \text{query \& evidence holds}}{\#\ \text{evidence holds}}$$

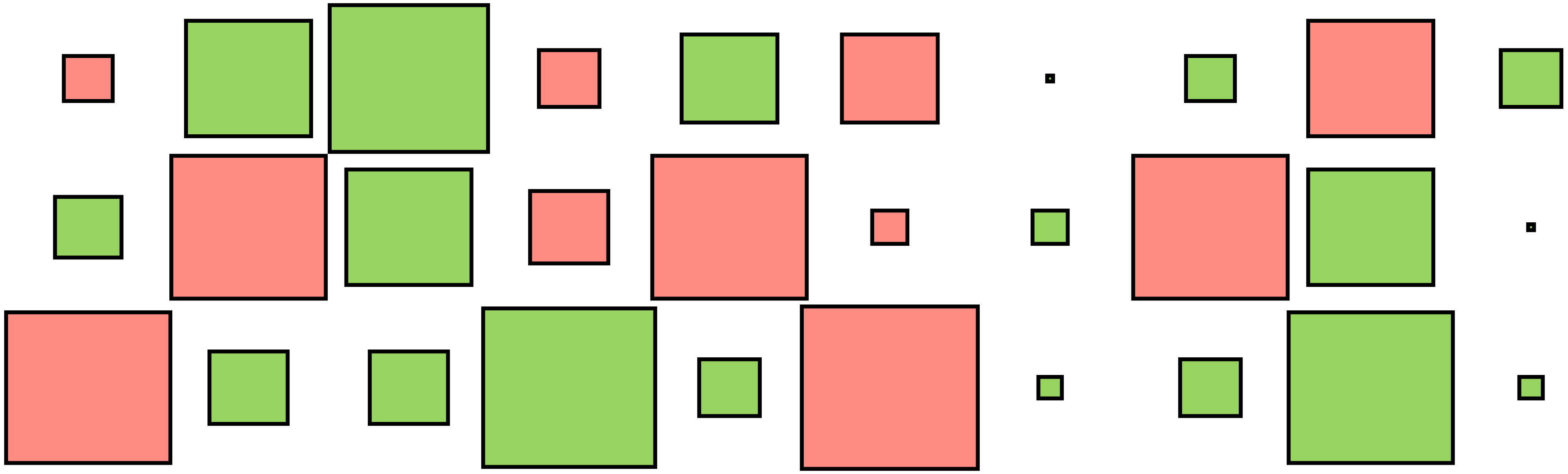
Likelihood Weighting

- P(query | evidence) ?



Likelihood Weighting

- P(query | evidence) ?



LW for DC

Given a goal G and the global variables $w_q^{(i)}, iq, x^{P(i)}$, applying a rule produces a new goal G' and modifies the global variables:

1. G' is the new goal obtained from G using a kind of SLD-resolution step;
2. if a new variable r is sampled with value v,
   – set $w_q^{(i)} \leftarrow w_q^{(i)} \frac{p(r=v|x^{P(i)})}{g(r=v|x^{P(i)})}$ (based on LW) and
   – $x^{P(i)} \leftarrow x^{P(i)} \cup \{r = v\}$.
   In addition, if $r = Val \in iq$ with r grounded, then:
   – $iq \leftarrow iq\theta$ with $\theta = \{Val = v\}$.
3. if a new atom a is proved set $x^{P(i)} \leftarrow x^{P(i)} \cup \{a\}$.

$$1:(\mathtt{color}(2)\sim=\mathtt{black});\, w_q^{(i)}=1;\, x^{P(i)}=\emptyset$$

$\downarrow$ 2b on (7) :

$$2:(\mathtt{material}(2)\sim=\mathtt{metal}, \mathtt{sample}(\mathtt{color}(2), \mathcal{D}_{\mathtt{color}(2)}), \mathtt{color}(2)\sim=\mathtt{black});\, w_q^{(i)}=1;\, x^{P(i)}=\emptyset$$

$\downarrow$ 2b on (9) :

$$3:(\mathtt{n}\sim=\mathtt{N}, \mathtt{between}(1,\mathtt{N},2), \mathtt{sample}(\mathtt{material}(2), \mathcal{D}_{\mathtt{material}(2)}), \mathtt{material}(2)\sim=\mathtt{metal},$$
$$\mathtt{sample}(\mathtt{color}(2), \mathcal{D}_{\mathtt{color}(2)}), \mathtt{color}(2)\sim=\mathtt{black});\, w_q^{(i)}=1;\; x^{P(i)}=\emptyset$$

$\downarrow$ 2b on (6) :

$$4:(\mathtt{sample}(\mathtt{n}, \mathcal{D}_{\mathtt{n}}), \mathtt{n}\sim=\mathtt{N}, \mathtt{between}(1, \simeq(\mathtt{n}), 2), \mathtt{sample}(\mathtt{material}(2), \mathcal{D}_{\mathtt{material}(2)}),$$
$$\mathtt{material}(2)\sim=\mathtt{metal}, \mathtt{sample}(\mathtt{color}(2), \mathcal{D}_{\mathtt{color}(2)}), \mathtt{color}(2)\sim=\mathtt{black});\, w_q^{(i)}=1;\, x^{P(i)}=\emptyset$$

$\downarrow$ 3b :

$$5:(\mathtt{n}\sim=3, \mathtt{between}(1,3,2), \mathtt{sample}(\mathtt{material}(2), \mathcal{D}_{\mathtt{material}(2)}), \mathtt{material}(2)\sim=\mathtt{metal},$$
$$\mathtt{sample}(\mathtt{color}(2), \mathcal{D}_{\mathtt{color}(2)}), \mathtt{color}(2)\sim=\mathtt{black});\, w_q^{(i)}=1;\; x^{P(i)}=\{\mathtt{n}=3\}$$

$\downarrow$ 2a followed by 1a

$$6:(\mathtt{sample}(\mathtt{material}(2), \mathcal{D}_{\mathtt{material}(2)}), \mathtt{material}(2)\sim=\mathtt{metal}, \mathtt{sample}(\mathtt{color}(2),$$
$$\mathtt{color}(2)\sim=\mathtt{black});\, w_q^{(i)}=1;\; x^{P(i)}=\{\mathtt{n}=3\}$$

$\downarrow$ 3b :

$$7:(\mathtt{material}(2)\sim=\mathtt{metal}, \mathtt{sample}(\mathtt{color}(2), \mathcal{D}_{\mathtt{color}(2)}), \mathtt{color}(2)\sim=\mathtt{black})$$
$$w_q^{(i)}=1;\; x^{P(i)}=\{\mathtt{n}=3, \mathtt{material}(2)=\mathtt{wood}\}$$

fail, backtracking to 1

$$\mathtt{n} \sim \mathtt{uniform}([1,2,3,4,5,6,7,8,9,10]). \tag{6}$$

$$\mathtt{color}(\mathtt{X}) \sim \mathtt{uniform}([\mathtt{grey},\mathtt{blue},\mathtt{black}]) \leftarrow \mathtt{material}(\mathtt{X}) \sim= \mathtt{metal}. \tag{7}$$

$$\mathtt{color}(\mathtt{X}) \sim \mathtt{uniform}([\mathtt{black},\mathtt{brown}]) \leftarrow \mathtt{material}(\mathtt{X}) \sim= \mathtt{wood}. \tag{8}$$

$$\mathtt{material}(\mathtt{X}) \sim \mathtt{finite}([0.3:\mathtt{wood}, 0.7:\mathtt{metal}]) \leftarrow \mathtt{n} \sim= \mathtt{N}, \mathtt{between}(1,\mathtt{N},\mathtt{X}). \tag{9}$$

$$\mathtt{drawn}(\mathtt{Y}) \sim \mathtt{uniform}(\mathtt{L}) \leftarrow \mathtt{n} \sim= \mathtt{N}, \mathtt{findall}(\mathtt{X}, \mathtt{between}(1,\mathtt{N},\mathtt{X}), \mathtt{L}). \tag{10}$$

$$\mathtt{size}(\mathtt{X}) \sim \mathtt{beta}(2,3) \leftarrow \mathtt{material}(\mathtt{X}) \sim= \mathtt{metal}. \tag{11}$$

$$\mathtt{size}(\mathtt{X}) \sim \mathtt{beta}(4,2) \leftarrow \mathtt{material}(\mathtt{X}) \sim= \mathtt{wood}. \tag{12}$$

$\vdash 1: (\mathtt{color}(2) \sim= \mathtt{black}); w_q^{(i)} = 1; x^{P(i)} = \emptyset$

$\downarrow$ 2b on (8) :

$9: (\mathtt{material}(2) \sim= \mathtt{wood}, \mathtt{sample}(\mathtt{color}(2), \mathcal{D}_{\mathtt{color}(2)}), \mathtt{color}(2) \sim= \mathtt{black})$

$\quad w_q^{(i)} = 1;\ x^{P(i)} = \{\mathtt{n} = 3, \mathtt{material}(2) = \mathtt{wood}\}$

$\downarrow$ 2a :

$10: (\mathtt{sample}(\mathtt{color}(2), \mathcal{D}_{\mathtt{color}(2)}), \mathtt{color}(2) \sim= \mathtt{black}); w = 1; x^{P(i)} = \{\mathtt{n}=3, \mathtt{material}(2)=\mathtt{wood}\}$

$\downarrow$ 3a :

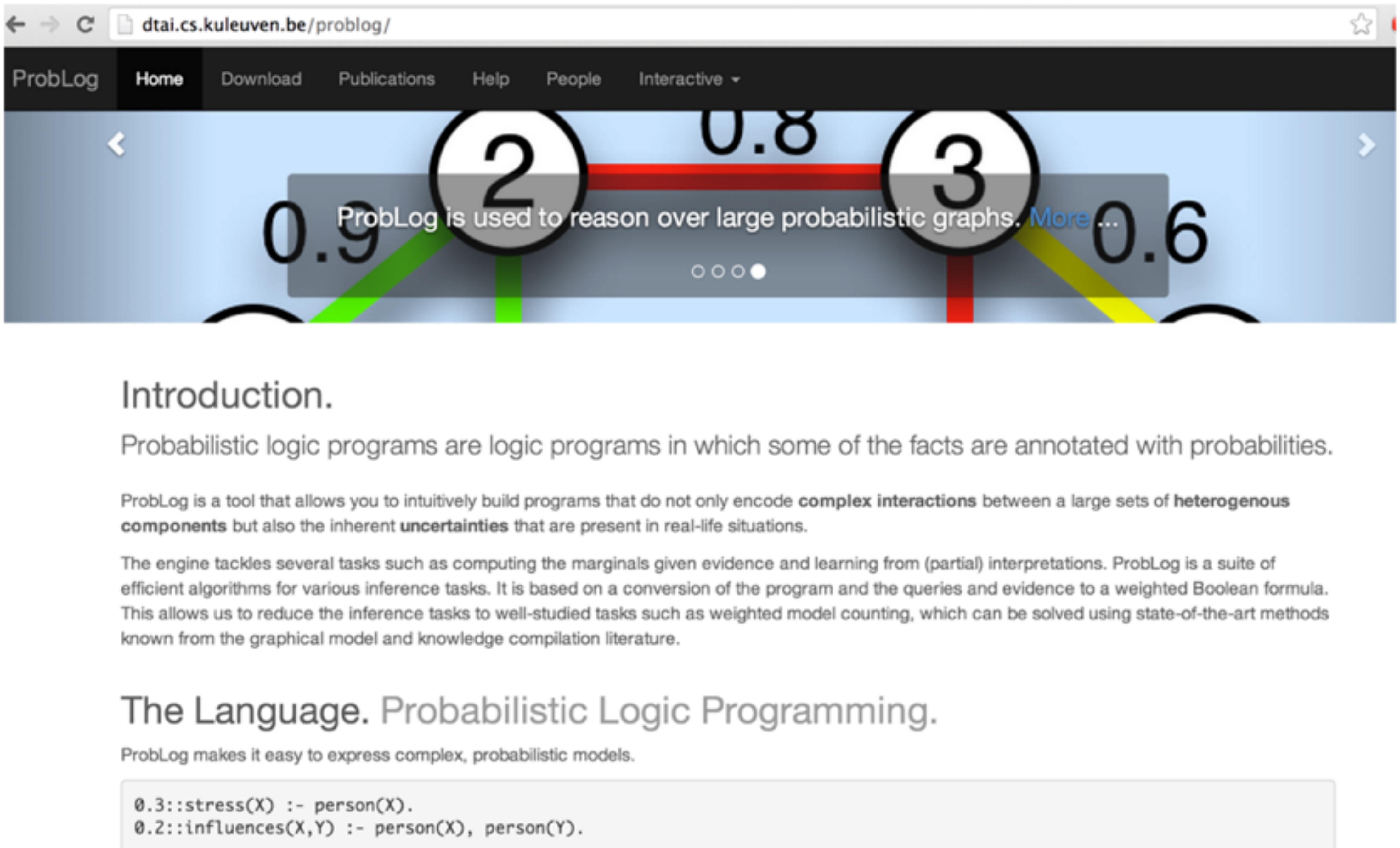
$11: (\mathtt{color}(2) \sim= \mathtt{black}); w_q^{(i)} = 1/3;\ x^{P(i)} = \{\mathtt{n} = 3, \mathtt{material}(2) = \mathtt{wood}, \mathtt{color}(2) = \mathtt{black}\}$

$\downarrow$ 1a :

$12: \Box; w_q^{(i)} = 1/3;\ x^{P(i)} = \{\mathtt{n} = 3, \mathtt{material}(2) = \mathtt{wood}, \mathtt{color}(2) = \mathtt{black}\}$

can cope with evidence like color(1) = color(2)
and size(1) = 0.356, size(1)=size(2), …
outperforms BLOG … unification + LW

http://dtai.cs.kuleuven.be/problog



PART III: Learning

Parameter Learning

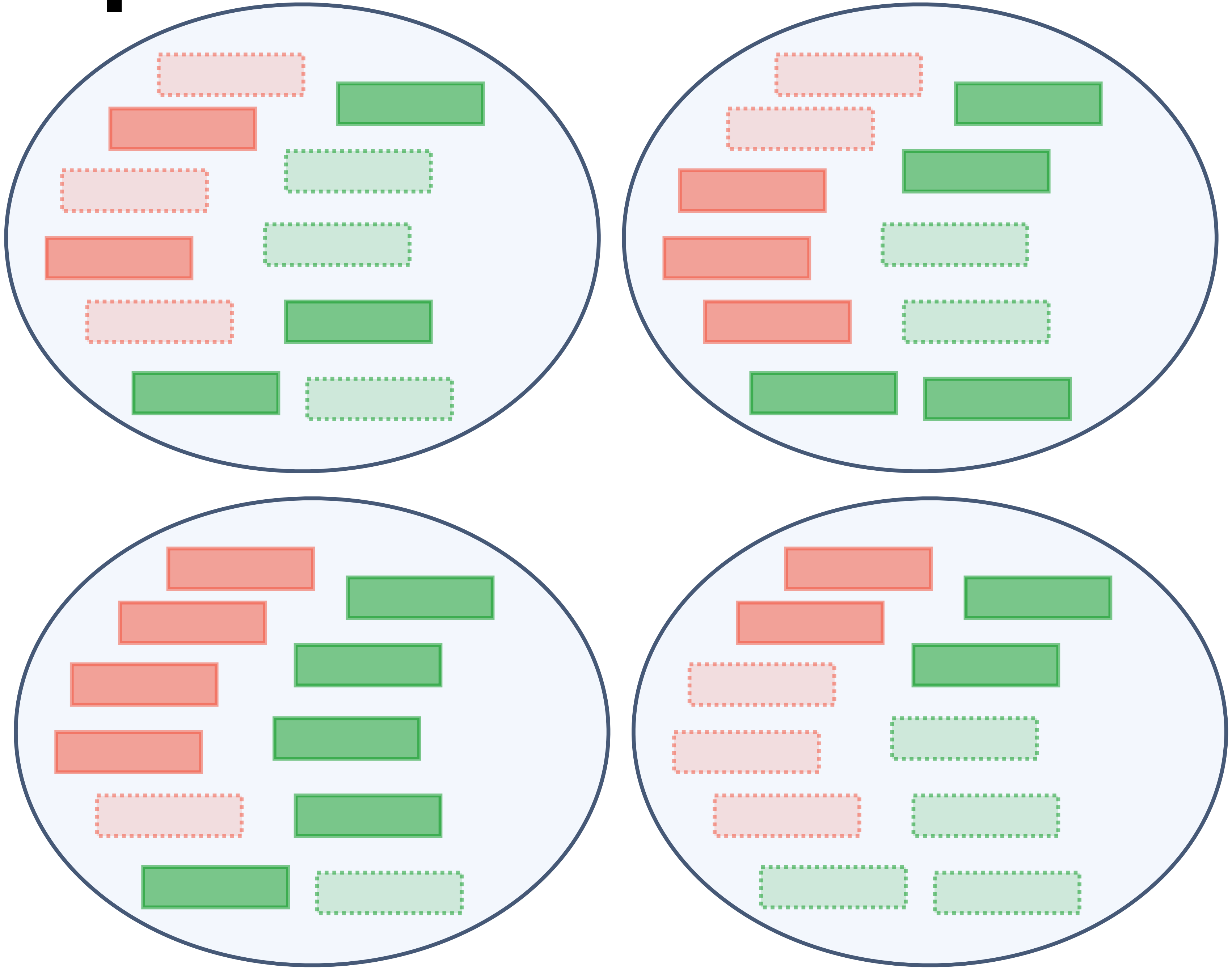
e.g., webpage classification model

for each *CLASS1*, *CLASS2* and each *WORD*

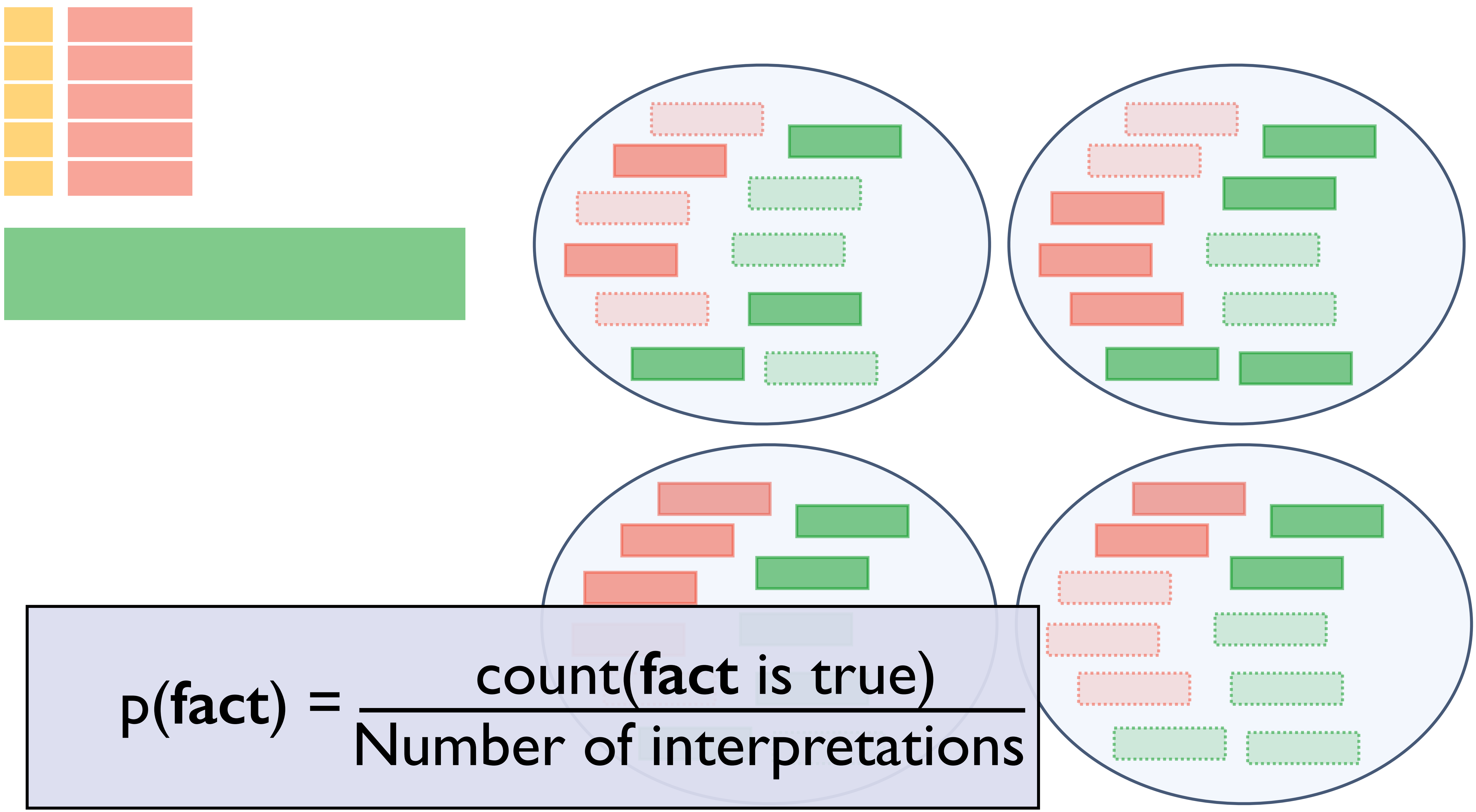
?? :: link_class(Source,Target,*CLASS1*,*CLASS2*).
?? :: word_class(*WORD*,*CLASS*).

```
class(Page,C)    :-   has_word(Page,W), word_class(W,C).

class(Page,C)          :-   links_to(OtherPage,Page),
    class(OtherPage,OtherClass),
                                link_class(OtherPage,Page,OtherClass,C).
```

Sampling Interpretations

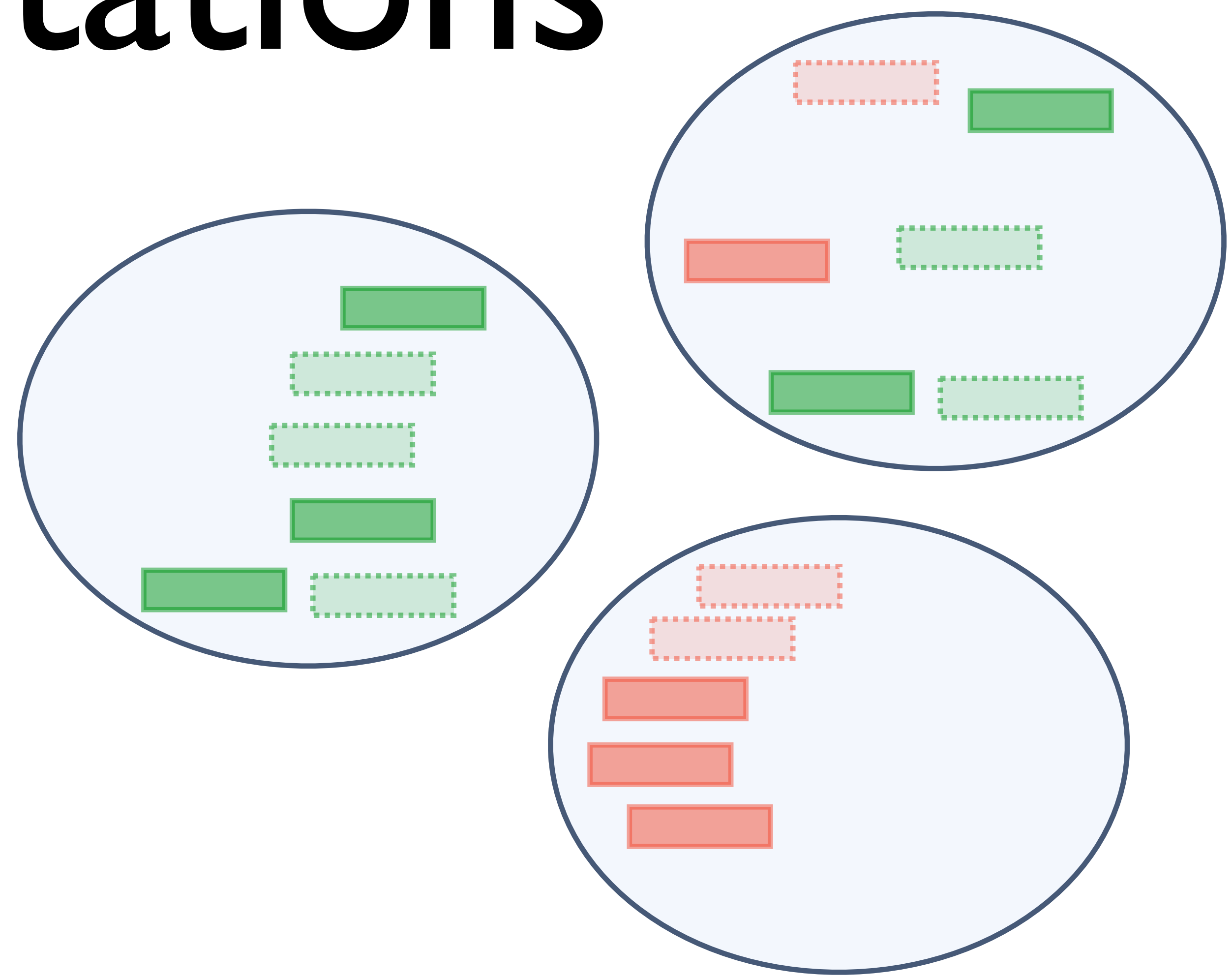


Parameter Estimation



$$p(\textbf{fact}) = \frac{\text{count}(\textbf{fact}\text{ is true})}{\text{Number of interpretations}}$$

Learning from partial interpretations



- Not all facts observed
- Soft-EM
- use **expected count** instead of **count**
- **P(Q |E) -- conditional queries !**

 [Gutmann et al, ECML 11; Fierens et al, TPLP 14]

Learning from single facts / entailment

- Only true facts are given; e.g. as in HMM
  - key setting in PRISM, also in ProbLog
- EM-based, variations exist
- use **expected count** instead of **count**
- **P(Q |E) -- conditional queries !**

[Gutmann et al, ECML 08; Sato 95]

Bayesian Parameter Learning

- Learning as inference (e.g., Church)
- Prior distributions for parameters
- Given data, find most likely parameter values

Information Extraction in NELL

Recently-Learned Facts twitter

Refresh

instance	iteration	date learned	confidence
kelly_andrews is a female	826	29-mar-2014	98.7
investment_next_year is an economic sector	829	10-apr-2014	95.3
shibenik is a geopolitical entity that is an organization	829	10-apr-2014	97.2
quality_web_design_work is a character trait	826	29-mar-2014	91.0
mercedes_benz_cls_by_carlsson is an automobile manufacturer	829	10-apr-2014	95.2
social_work is an academic program at the university rutgers_university	827	02-apr-2014	93.8
dante wrote the book the_divine_comedy	826	29-mar-2014	93.8
willie_aames was born in the city los_angeles	831	16-apr-2014	100.0
kitt_peak is a mountain in the state or province arizona	831	16-apr-2014	96.9
greenwich is a park in the city london	831	16-apr-2014	100.0

instances for many different relations

degree of certainty

NELL: http://rtw.ml.cmu.edu/rtw/

Rule learning in NELL

- Original approach
  - Make probabilistic data deterministic
  - run classic rule-learner (variant of FOIL)
  - re-introduce probabilities on learned rules and predict

Probabilistic Rule Learning

- Learn the rules directly in a PLP setting
- Generalize relational learning and inductive logic programming directly towards probabilistic setting
- Traditional rule learning/ILP as a special case
- Apply to probabilistic databases like NELL
- Approach in PPR (Cohen et al) and in ProbLog (IJCAI-15)

NELL

Table 5: Number of facts per predicate (NELL athlete dataset)

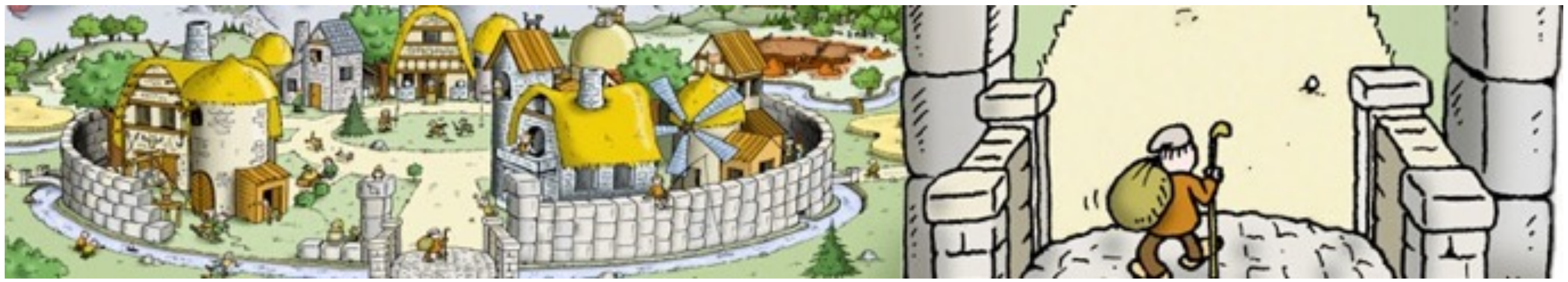
athletecoach(person,person)	18	athleteplaysforteam(person,team)	721
athleteplayssport(person,sport)	1921	teamplaysinleague(team,league)	1085
athleteplaysinleague(person,league)	872	athletealsoknownas(person,name)	17
coachesinleague(person,league)	93	coachesteam(person,team)	132
teamhomestadium(team,stadium)	198	teamplayssport(team,sport)	359
athleteplayssportsteamposition(person,position)	255	athletehomestadium(person,stadium)	187
athlete(person)	1909	attraction(stadium)	2
coach(person)	624	female(person)	2
male(person)	7	hobby(sport)	5
organization(league)	1	person(person)	2
personafrica(person)	1	personasia(person)	4
personaustralia(person)	22	personcanada(person)	1
personeurope(person)	1	personmexico(person)	108
personus(person)	6	sport(sport)	36
sportsleague(league)	18	sportsteam(team)	1330
sportsteamposition(position)	22	stadiumoreventvenue(stadium)	171

athleteplaysforteam

```
athleteplaysforteam(A,B) :- coachesteam(A,B).
0.875::athleteplaysforteam(A,B) :- teamhomestadium(B,C), athletehomestadium(A,C).
0.99080::athleteplaysforteam(A,B) :- teamhomestadium(B,_), male(A), athleteplayssport(A,_).
0.75::athleteplaysforteam(A,B) :- teamhomestadium(B,_), athleteplaysinleague(A,C), teamplaysinleague(B,C),
 athlete(A).
0.75::athleteplaysforteam(A,B) :- teamplayssport(B,C), athleteplayssport(A,C), coach(A), teamplaysinleague(B,_).
0.97555::athleteplaysforteam(A,B) :- personus(A), teamplayssport(B,_).
0.762::athleteplaysforteam(A,B) :- teamplayssport(B,C), athleteplayssport(A,C), personmexico(A),
 teamplaysinleague(B,_).
0.52571::athleteplaysforteam(A,B) :- teamplayssport(B,C), athleteplayssport(A,C), athleteplaysinleague(A,_),
 teamplaysinleague(B,_), athlete(A), teamplayssport(B,C).
0.50546::athleteplaysforteam(A,B) :- teamplayssport(B,_), teamplaysinleague(B,C), athleteplaysinleague(A,C),
 athleteplayssport(A,_).
0.50::athleteplaysforteam(A,B) :- teamplayssport(B,_), teamplaysinleague(B,C), athleteplaysinleague(A,C).
0.52941::athleteplaysforteam(A,B) :- teamplayssport(B,_), teamhomestadium(B,_), coach(A), teamplaysinleague(B,_).
0.55287::athleteplaysforteam(A,B) :- teamplayssport(B,_), teamplaysinleague(B,C), athleteplaysinleague(A,C),
 athlete(A).
0.46875::athleteplaysforteam(A,B) :- teamplayssport(B,_), teamplaysinleague(B,_), coach(A),
 teamhomestadium(B,_).
```

Part IV: Dynamics

Dynamics: Evolving Networks



- *Travian*: A massively multiplayer real-time strategy game
  - Commercial game run by TravianGames GmbH
  - ~3.000.000 players spread over different "worlds"
  - ~25.000 players in one world

[Thon et al., MLJ 11, ECML 08]

World Dynamics

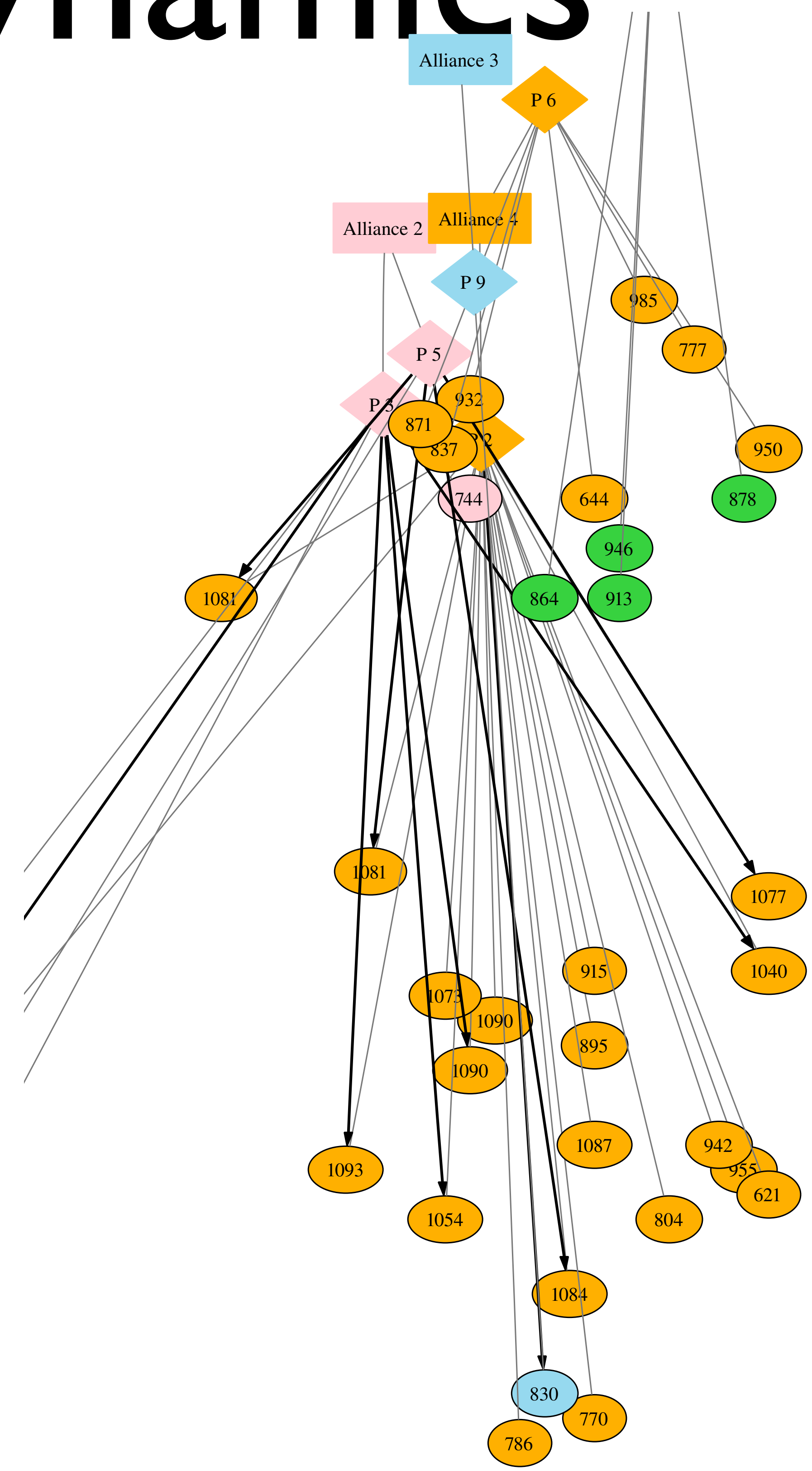
Fragment of world with

~10 alliances
~200 players
~600 cities

alliances color-coded

Can we build a model
of this world ?
Can we use it for playing
better ?

[Thon, Landwehr, De Raedt, ECML08]



World Dynamics

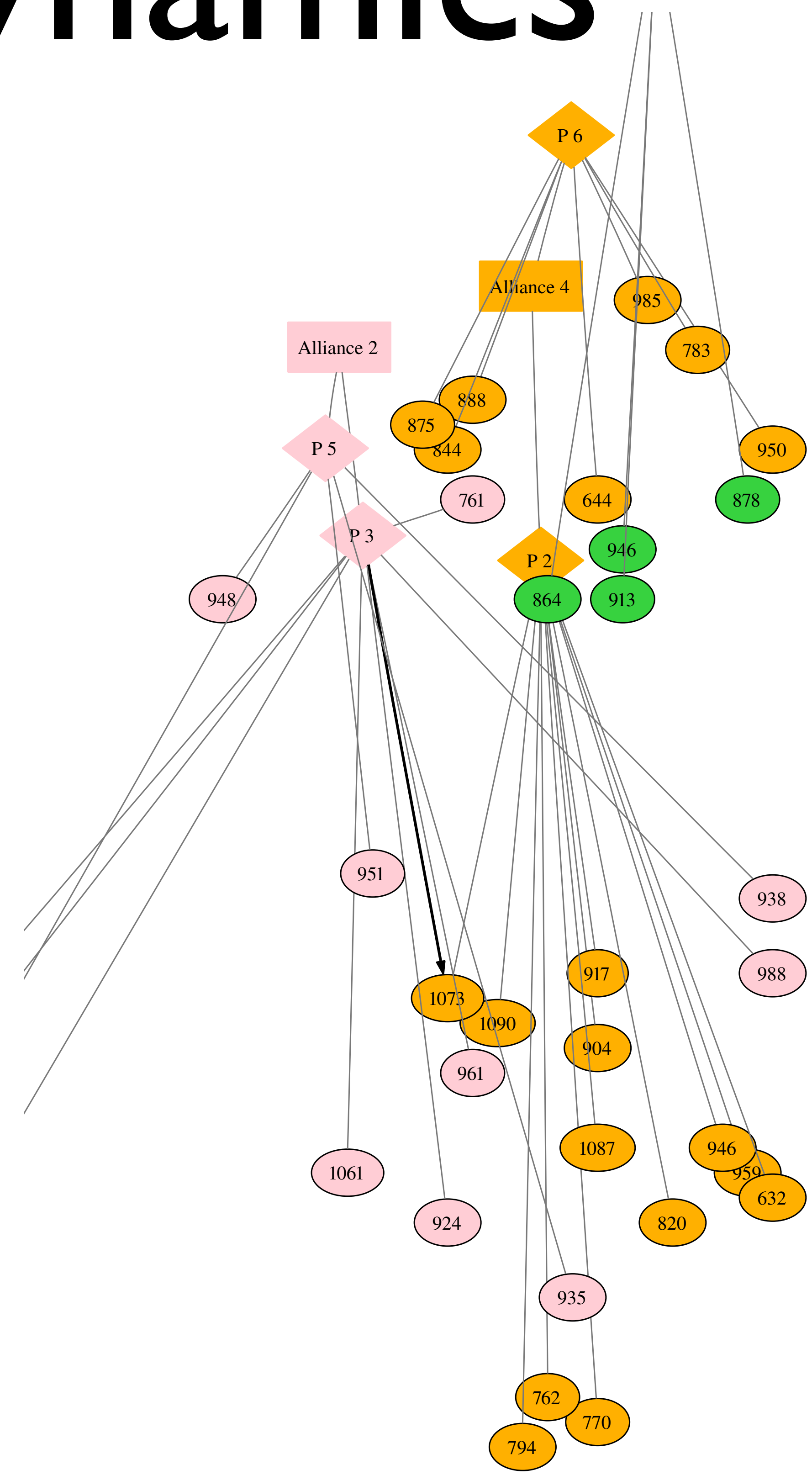
Fragment of world with

~10 alliances
~200 players
~600 cities

alliances color-coded

Can we build a model
of this world ?
Can we use it for playing
better ?

[Thon, Landwehr, De Raedt, ECML08]



World Dynamics

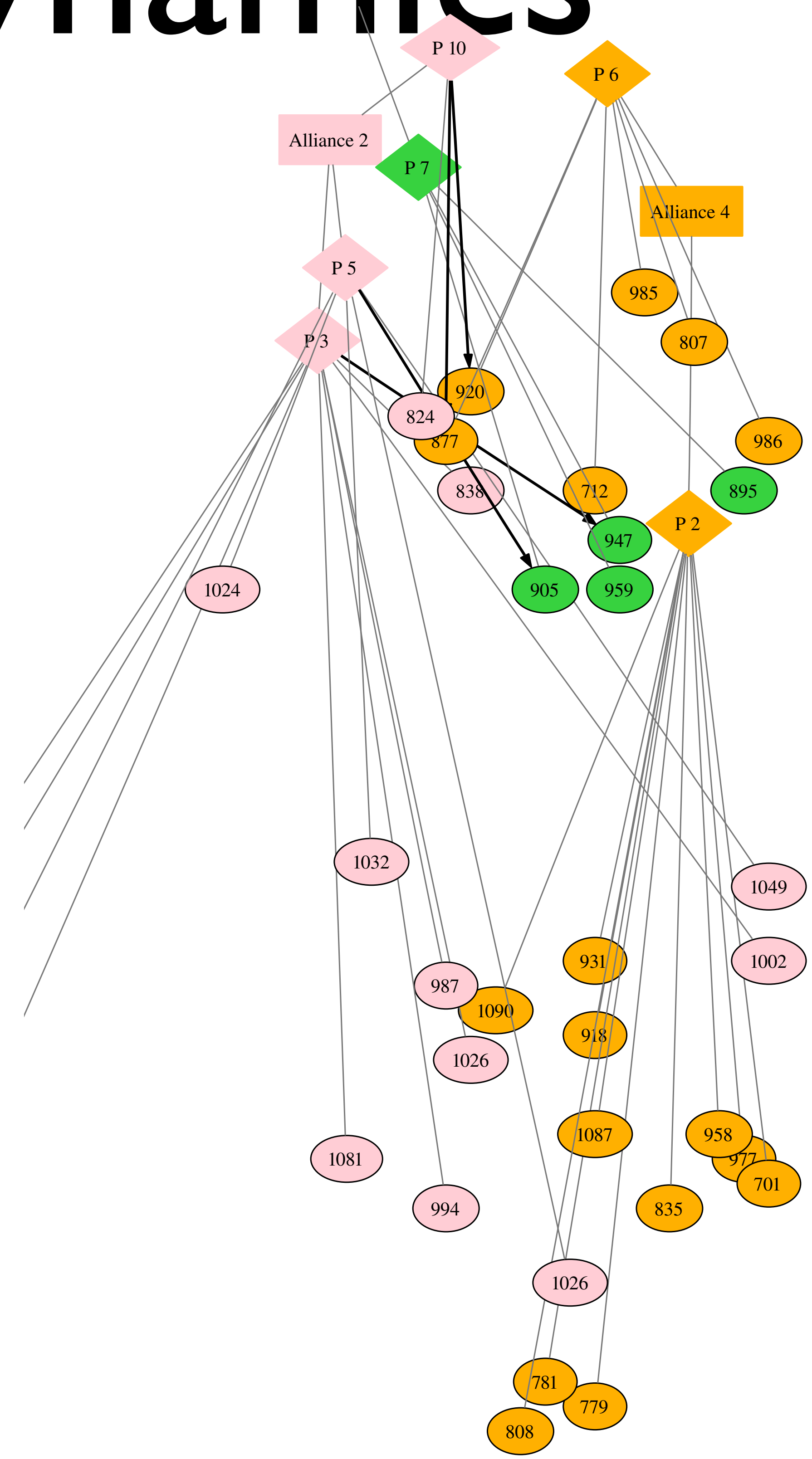
Fragment of world with

~10 alliances
~200 players
~600 cities

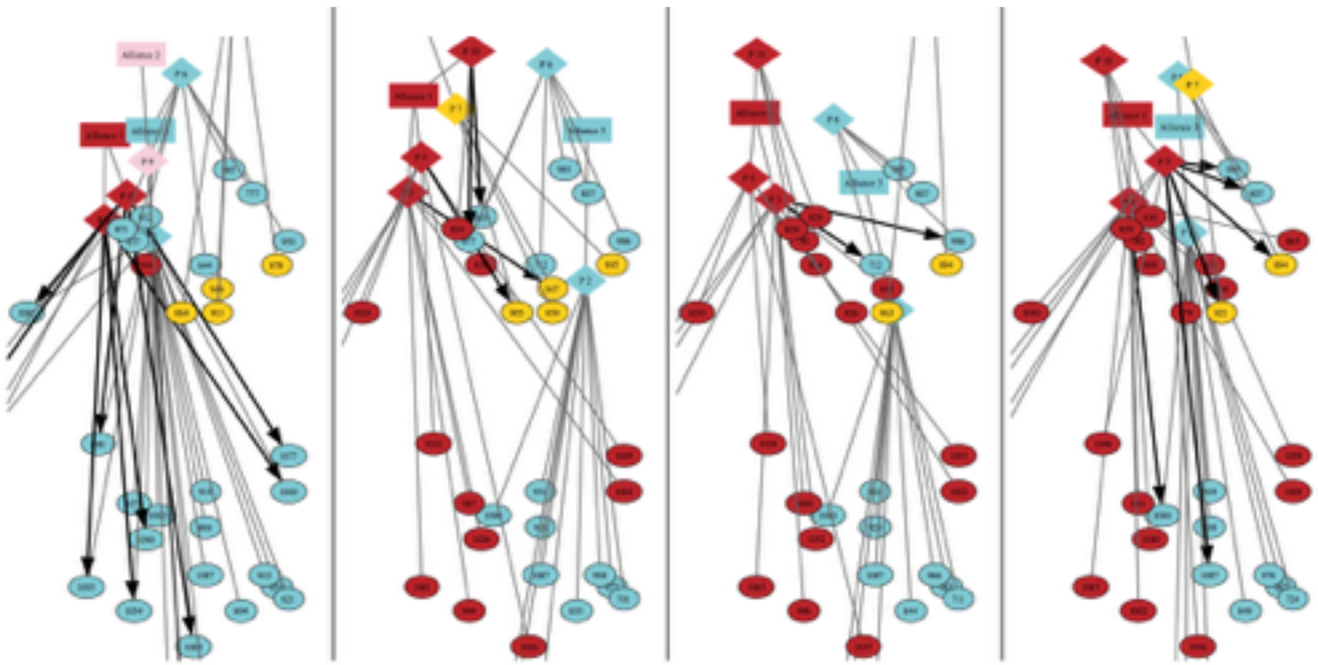
alliances color-coded

Can we build a model
of this world ?
Can we use it for playing
better ?

[Thon, Landwehr, De Raedt, ECML08]



Causal Probabilistic Time-Logic (CPT-L)



how does the world change over time?

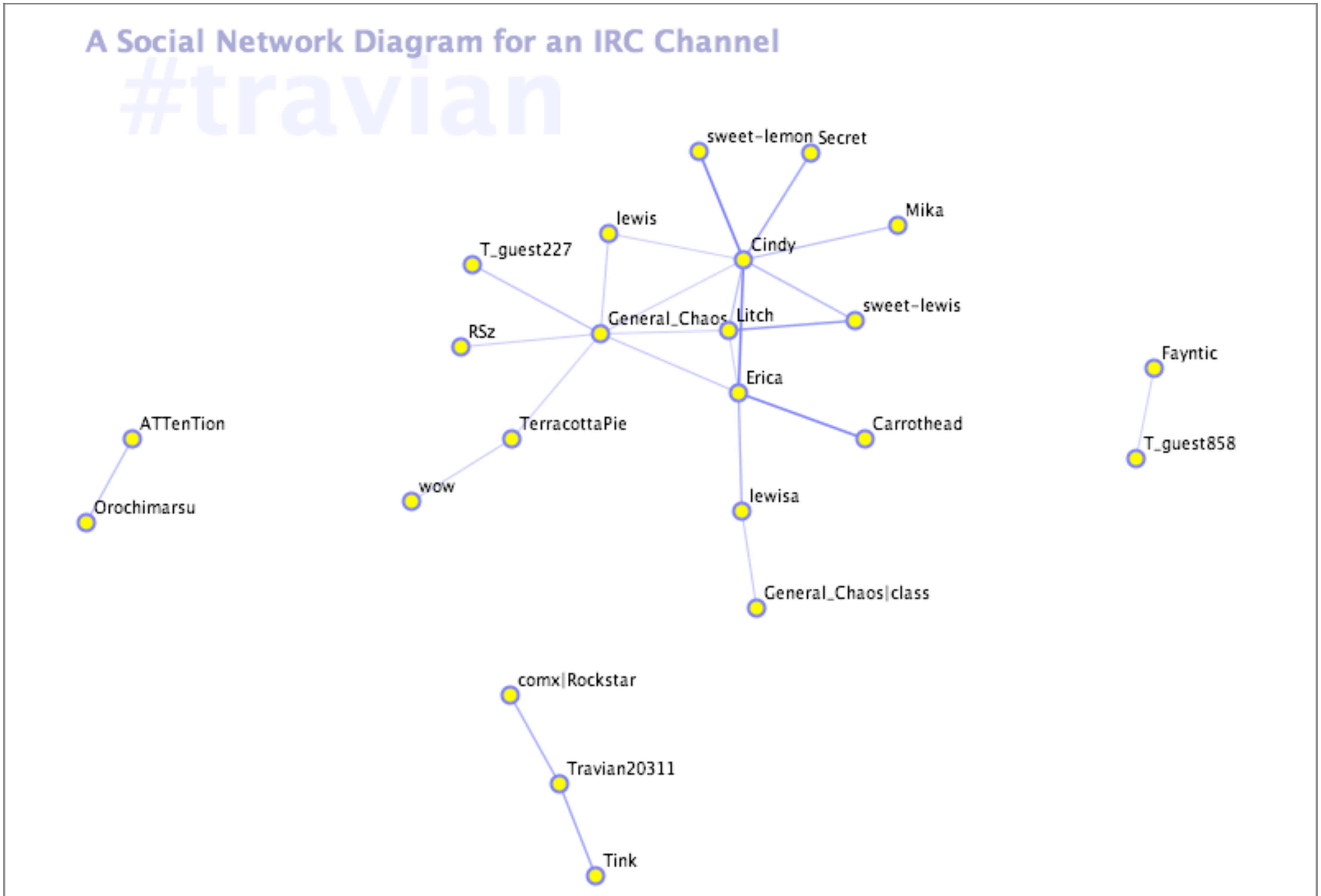
one of the **effects** holds at time T+1

```
0.4::conquest(Attacker,C); 0.6::nil <-
        city(C,Owner),city(C2,Attacker),close(C,C2).
```

if **cause** holds at time T

[Thon et al, MLJ 11]

Social Network of Chats



Limitations CPT-L

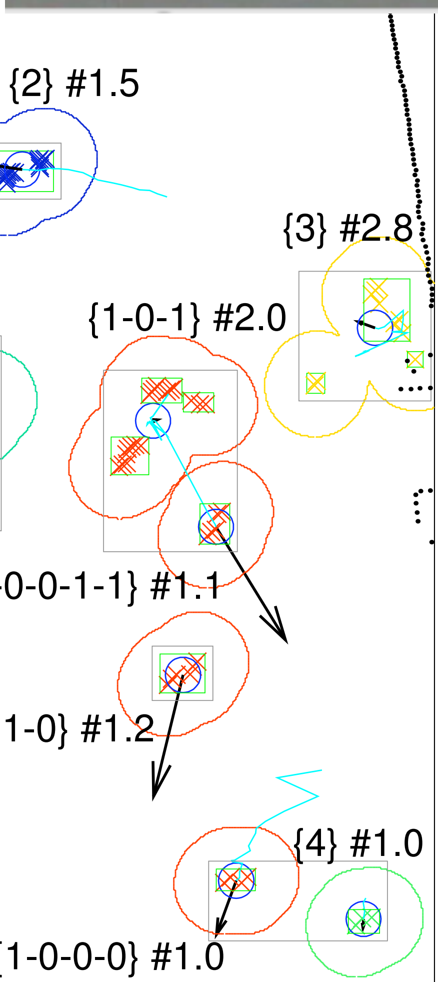
Inference slow / scalability

- uses knowledge compilation method
- compile formula for $P(I_{t+1}|I_{[0,t]})$
- exponential in number of time steps

No continuous distributions

- needed for robotics / relational tracking applications

Relational Tracking

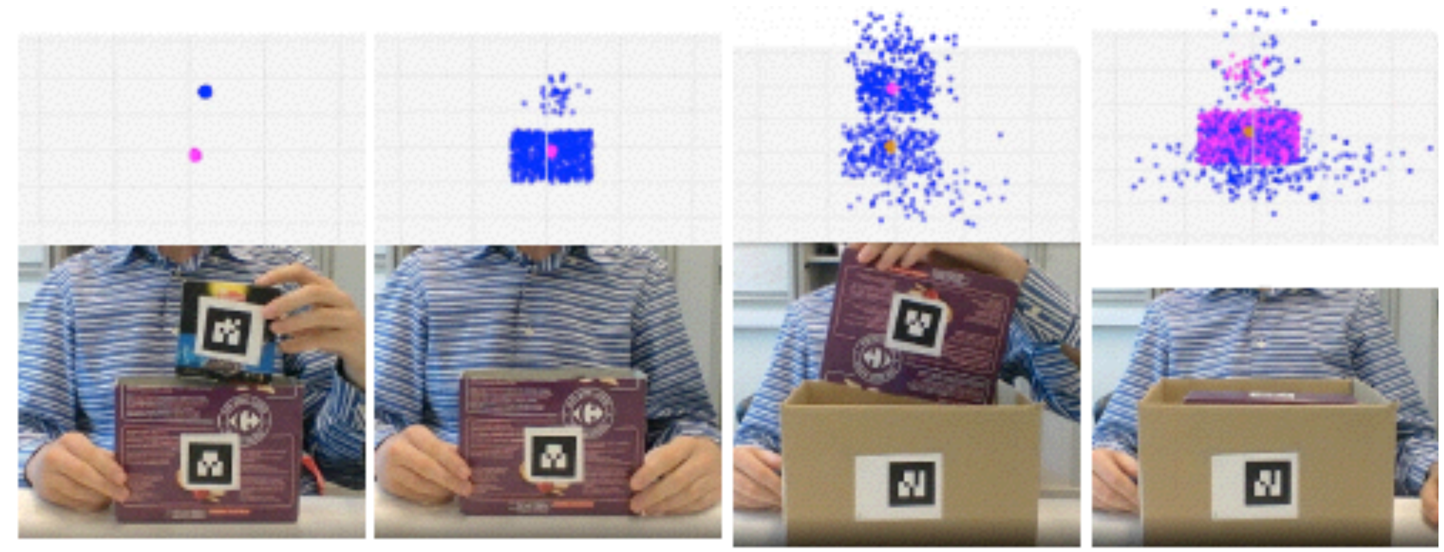


- Track people or objects over time? Even if temporarily hidden?
- Recognize activities?
- Infer object properties?

Relational State Estimation over Time

Magnetism scenario

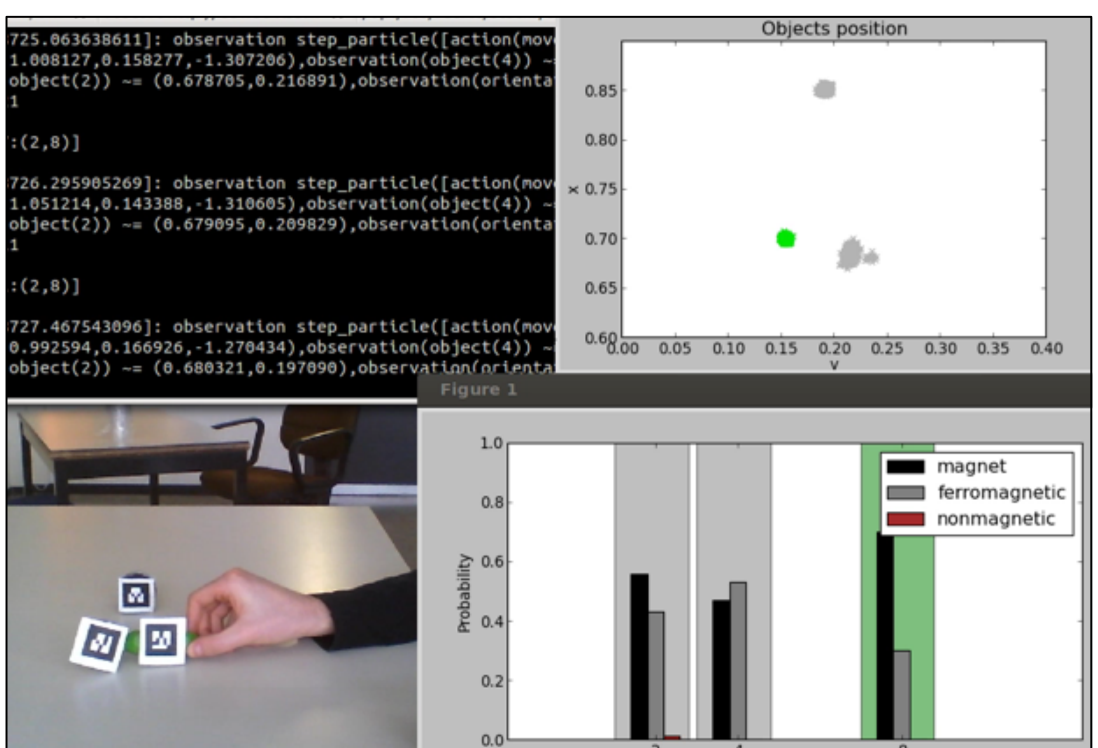
- object tracking
- category estimation from interactions



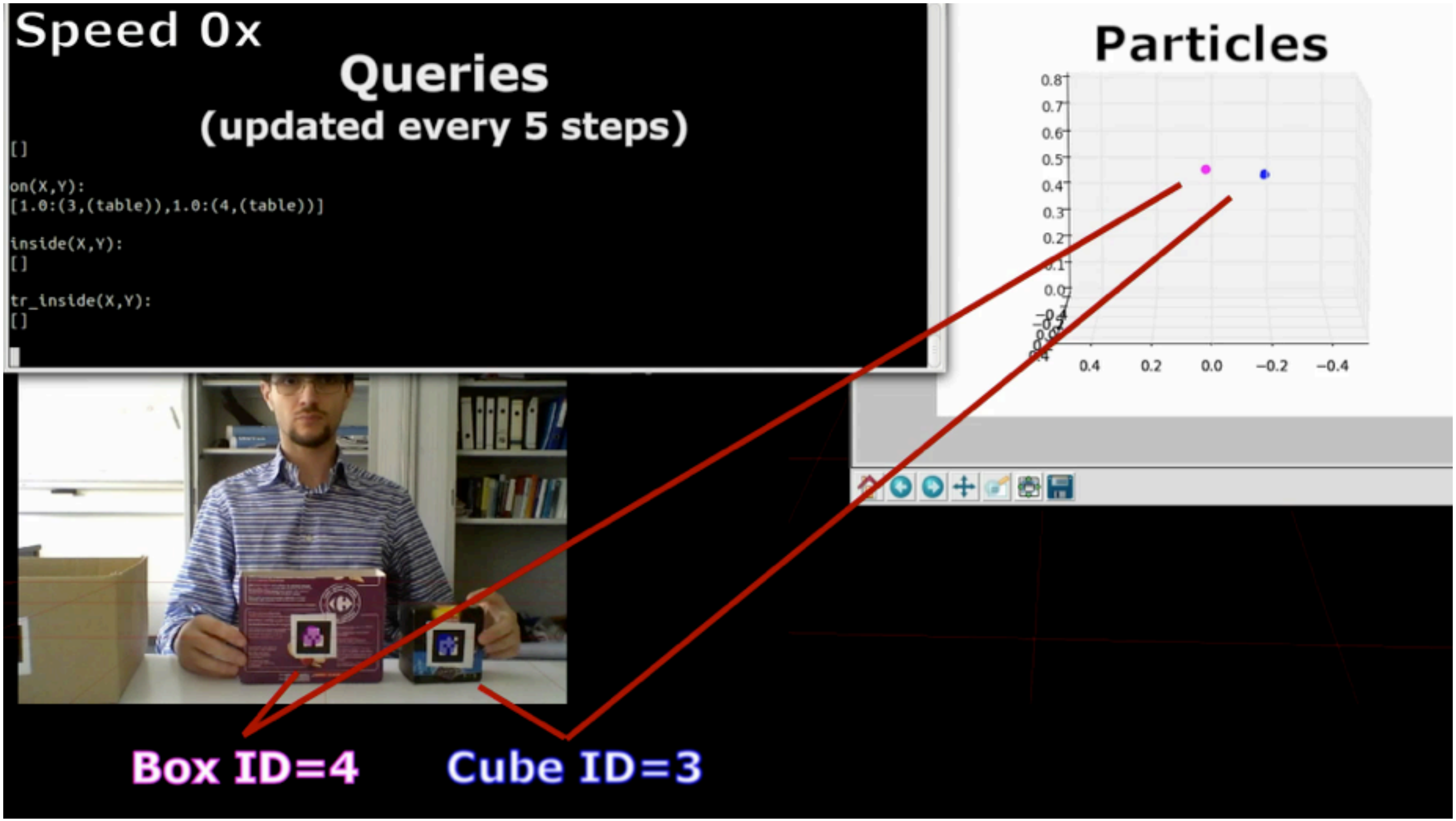
(a) cube on the box (b) cube inside the box (c) rotated box on a beige box (d) cube and box inside the beige box

Box scenario

- object tracking even when invisible
- estimate spatial relations



[Nitti et al, IROS 13]

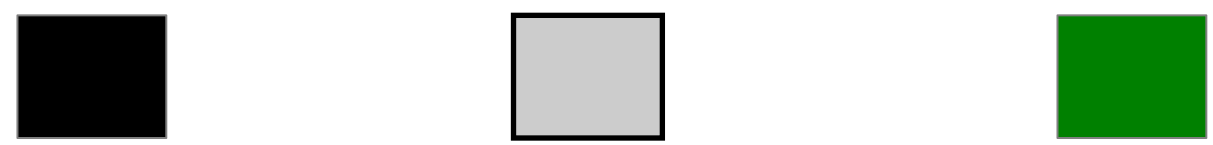


Speed 0x
Queries
(updated every 5 steps)
[]
on(X,Y):
[1.0:(3,(table)),1.0:(4,(table))]
inside(X,Y):
[]
tr_inside(X,Y):
[]
Particles
Box ID=4
Cube ID=3

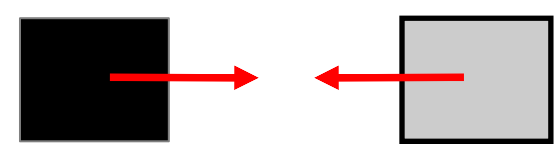
IROS 13

Magnetic scenario

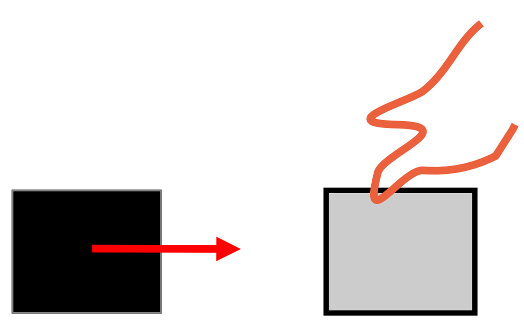
- 3 object types: magnetic, ferromagnetic, nonmagnetic

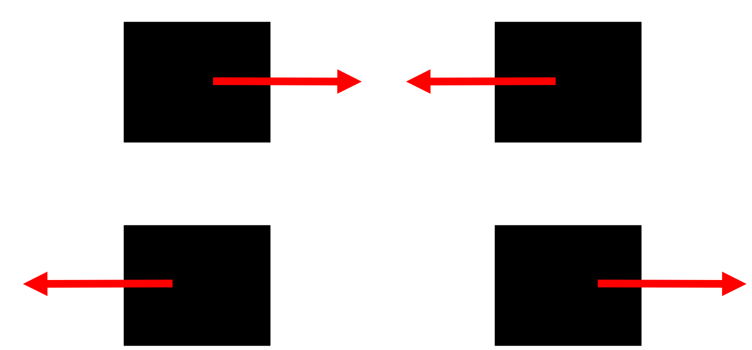


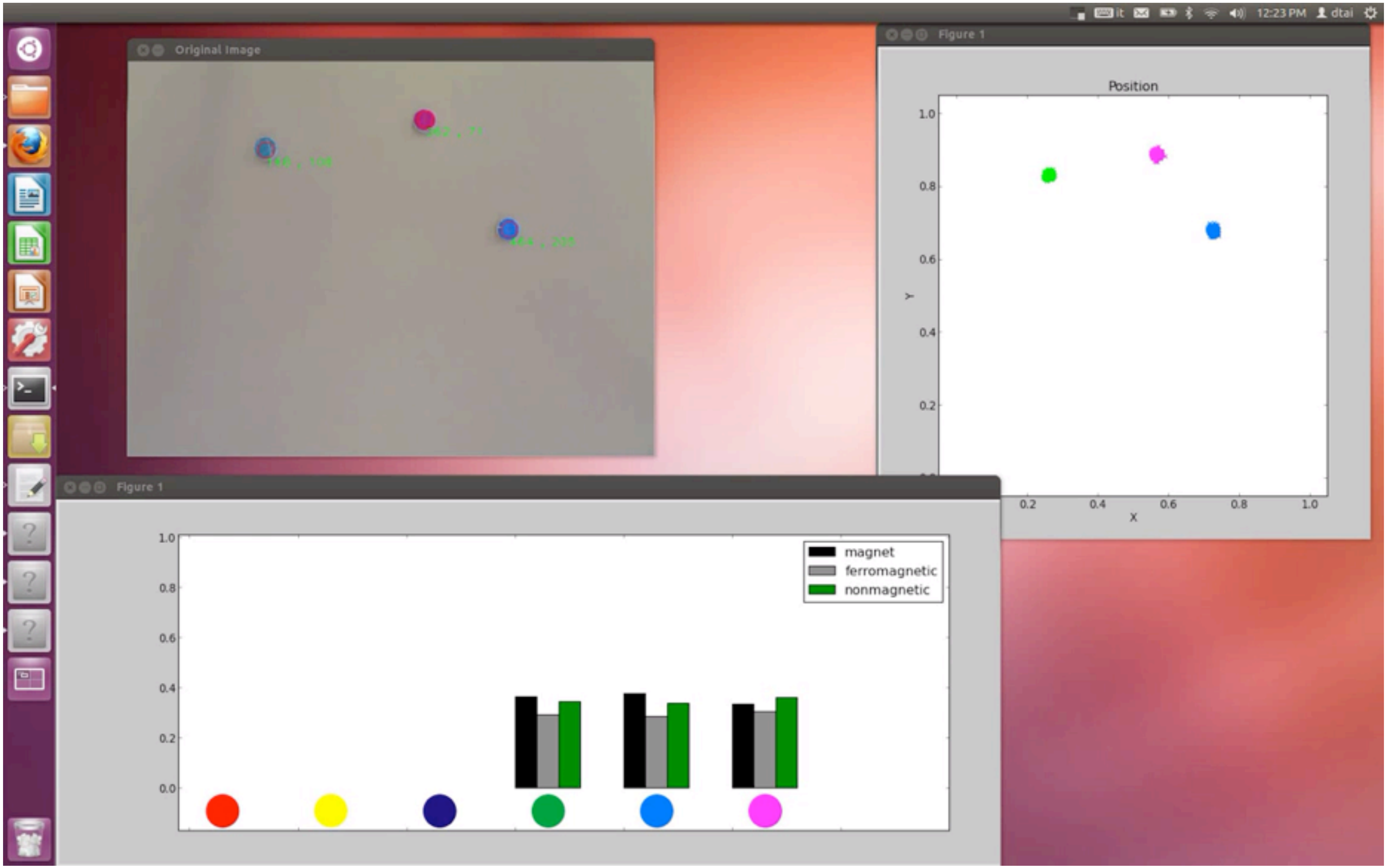
- Nonmagnetic objects do not interact
- A magnet and a ferromagnetic object attract each other



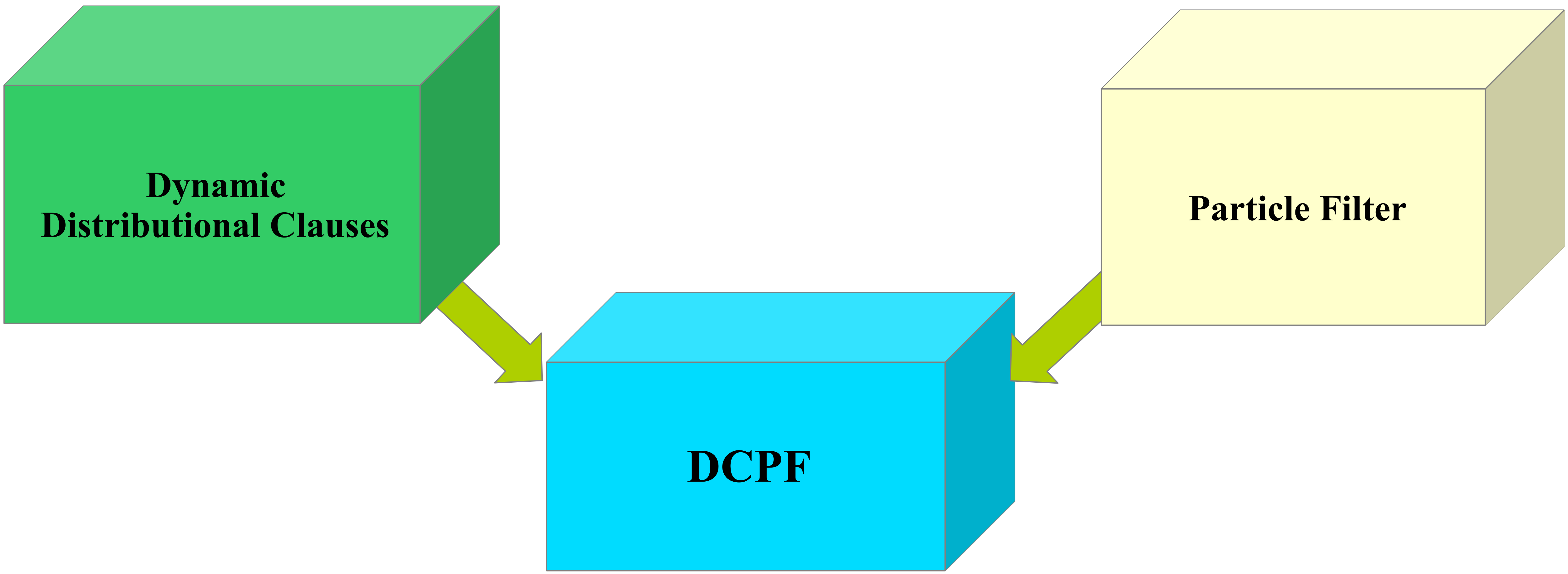
- Magnetic force that depends on the distance
- If an object is held magnetic force is compensated.






12:23 PM
dtai
Original Image
Figure 1
Position
1.0
0.8
0.6
0.4
0.2
Y
0.2
0.4
0.6
0.8
1.0
X
Figure 1
magnet
ferromagnetic
nonmagnetic
1.0
0.8
0.6
0.4
0.2
0.0

DC Particle Filter (DCPF)



Goal

Flexible (relational) state representation

Fast inference (state estimation) in general models

“A particle filter for hybrid relational domains” IROS 2013
D. Nitti, T. De Laet, L. De Raedt

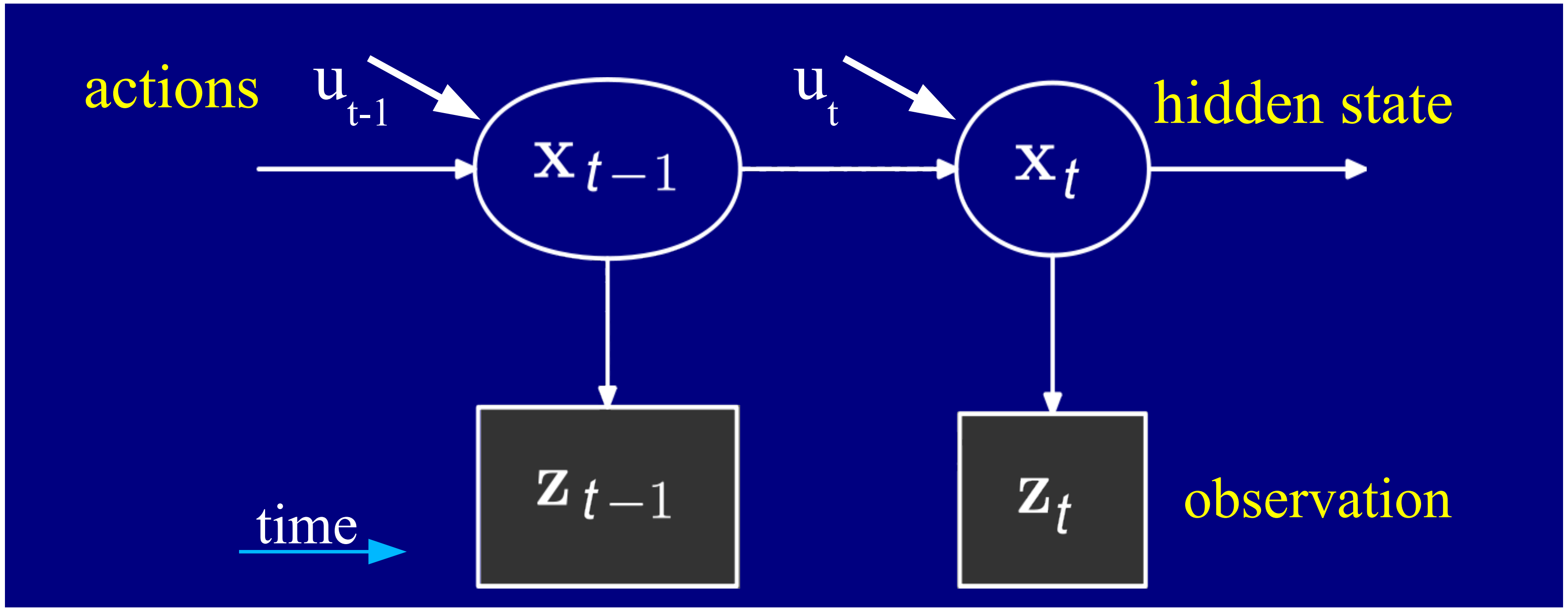
Dynamic Distributional Clauses

Prior distribution $p(x_0)$

State transition model $p(x_t|x_{t-1,}u_t)$

Measurement model $p(z_t|x_t)$

Other rules: $p(x'_t|x''_t)$



Magnetic scenario

- 3 object types: magnetic, ferromagnetic, nonmagnetic

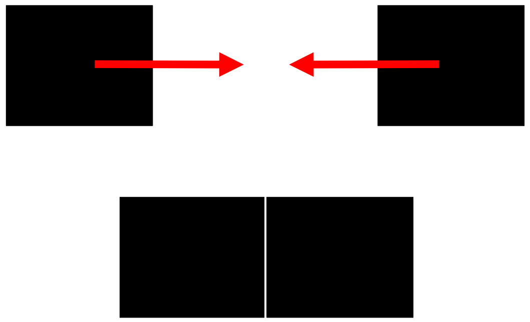
$\text{type}(X)_t \sim \text{finite}([1/3:\text{magnet},1/3:\text{ferromagnetic},1/3:\text{nonmagnetic}]) \leftarrow$
$\text{object}(X).$

- 2 magnets attract or repulse

$\text{interaction}(A,B)_t \sim \text{finite}([0.5:\text{attraction},0.5:\text{repulsion}]) \leftarrow$
$\text{object}(A), \text{object}(B), A<B, \text{type}(A)_t = \text{magnet}, \text{type}(B)_t = \text{magnet}.$

- Next position after attraction

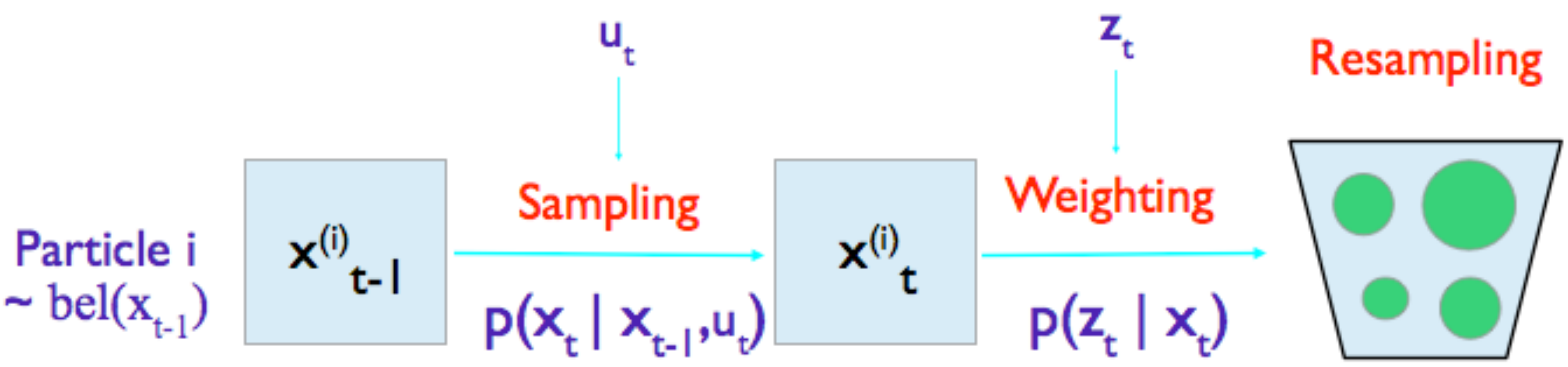
$\text{pos}(A)_{t+1} \sim \text{gaussian}(\text{middlepoint}(A,B)_t,\text{Cov}) \leftarrow$
$\text{near}(A,B)_t, \text{not}(\text{held}(A)), \text{not}(\text{held}(B)),$
$\text{interaction}(A,B)_t = \text{attr},$
$c/\text{dist}(A,B)_t^2 > \text{friction}(A)_t.$



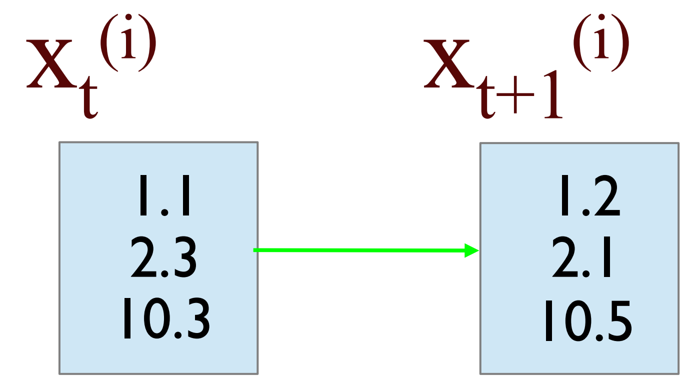
$\text{pos}(A)_{t+1} \sim \text{gaussian}(\text{pos}(A)_t,\text{Cov}) \leftarrow \text{not}(\ \text{attraction}(A,B)\ ).$

Particle Filter
(Sequential Monte Carlo)

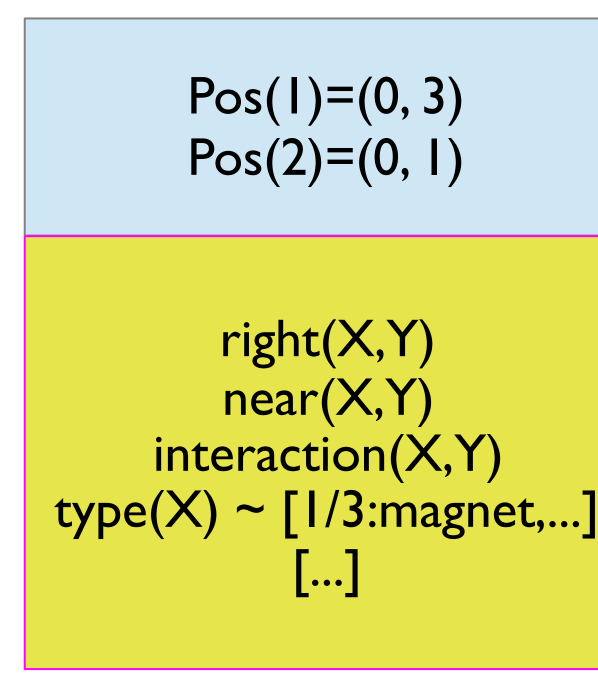
- Based on sampling → approximate inference
- Particles (samples) to represent bel(x_t)



Classical Particle Filter vs DCPF



- Classical PF
  - Fixed set of random variables
  - Update the entire state
- DCPF
  - **Adaptive state** (particle): the number of facts / random variables can change over time
  - Particles are partial interpretations
  - Expressive language



Optimized inference: partial state

Distributional Clauses Particle Filter (DCPF)

Sampled

Pos(1)=(0, 3)
Pos(2)=(0, 1)

Marginalized

right(X,Y)
near(X,Y)
interaction(X,Y)
type(X) ~ [1/3:magnet,...]
[...]

→

Pos(1)=(0, 2)
Pos(2)=(0, 1)
near(1,2)=true
type(1)=nonmagnetic

right(X,Y)
near(X,Y)
interaction(X,Y)
type(X) ~ [1/3:magnet,...]
[...]

Classical particle filter

Pos(1)=(0, 3)
Pos(2)=(0, 1)
near(1,2)=false
near(2,1)=false
interaction(1,2)=none
type(1)=nonmagnetic
type(2)=nonmagnetic
[...]

→

Pos(1)=(0, 2)
Pos(2)=(0, 1)
near(1,2)=true
near(2,1)=true
interaction(1,2)=none
type(1)=nonmagnetic
type(2)=nonmagnetic
[...]

Inference in DCPF

Two steps:

Query $p(z_{t+1}|x_{t+1})$ (weighting + part of sampling step)

Query $p(x_{t+1}|x_t, u_{t+1})$ (to complete the sampling step)

using the DC inference

particles are partial interpretations

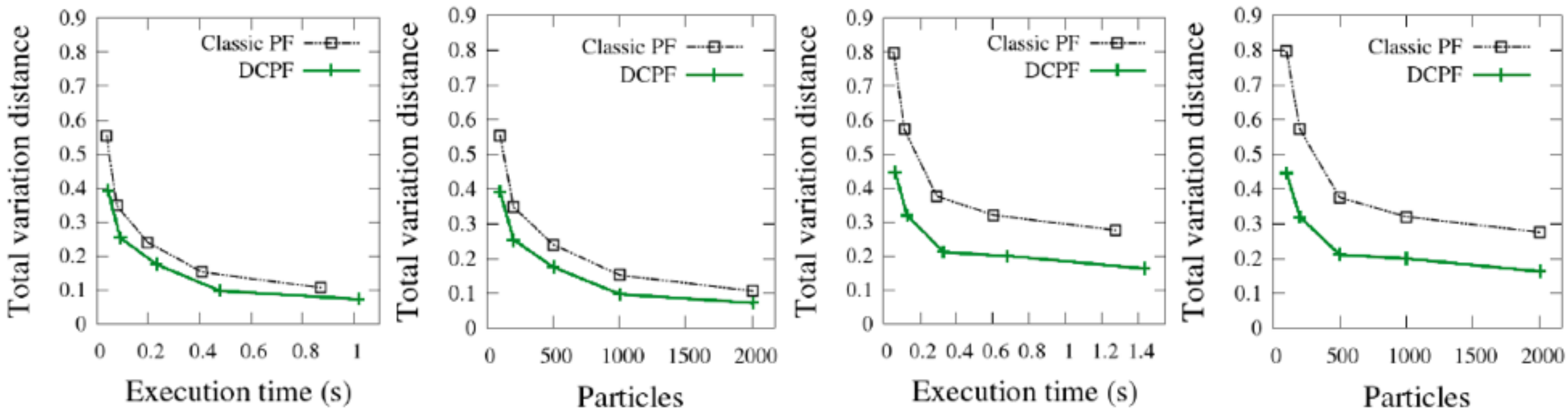
$bel(x_t)$ fully represented by $\{x_t^{(i)}\} \cup$ Program

History $\{x_{0:t-1}^{(i)}\}$ not necessary

Issue: particles (interpretations) may grow till becoming complete

Experiments

- Particles are partial state, remaining variables are marginalized
- Better performance in bigger models



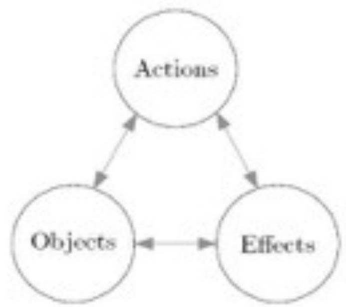
(a) 3 objects (b) 3 objects (c) 4 objects (d) 4 objects

Ongoing Work

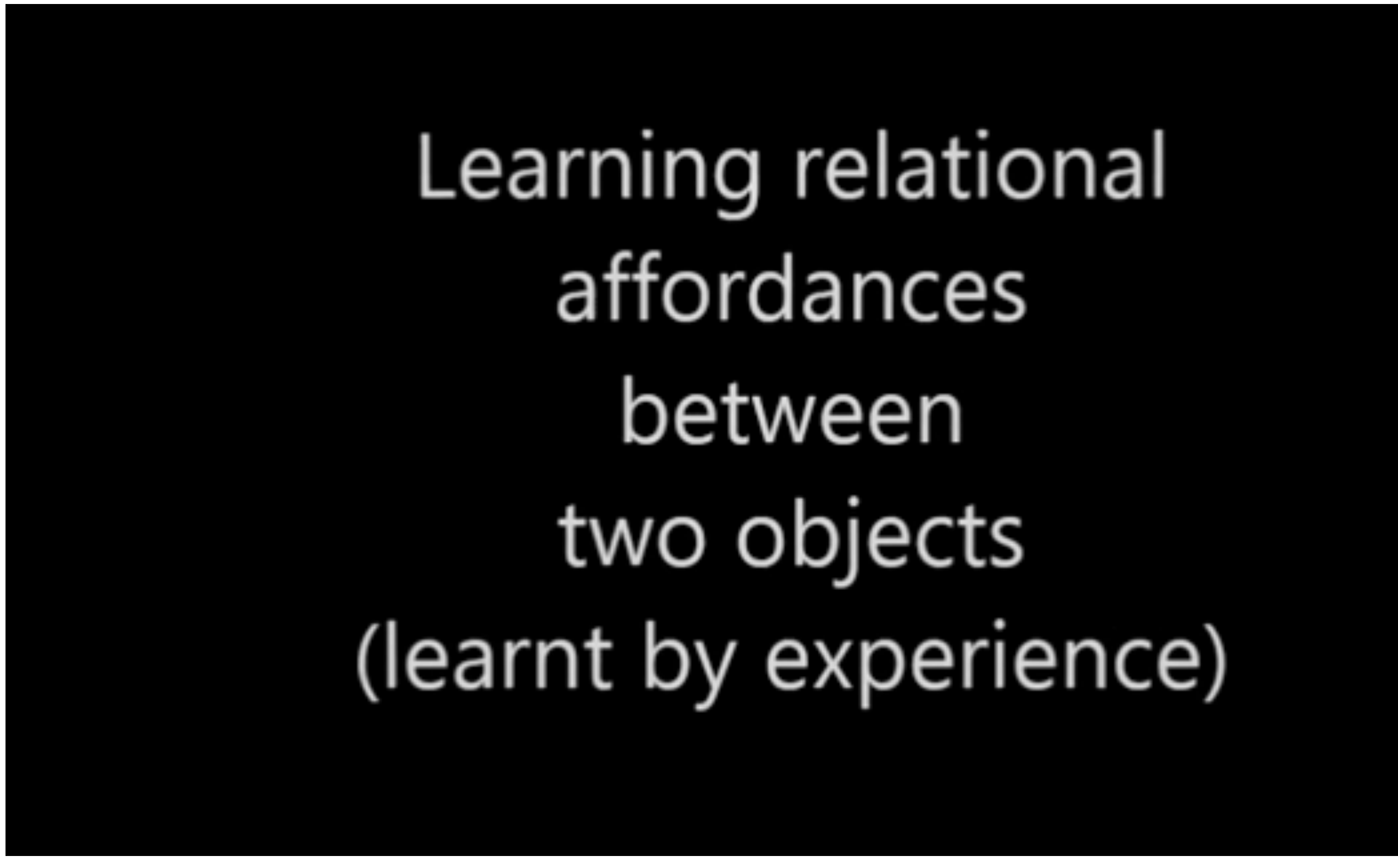
- Online parameter learning [Nitti, ICRA 2014]
- Integrate with planning [Nitti, ECML 15,EWRL 15]
- Applications in robotics (also to learn affordances)

Learning relational affordances

Learn probabilistic model



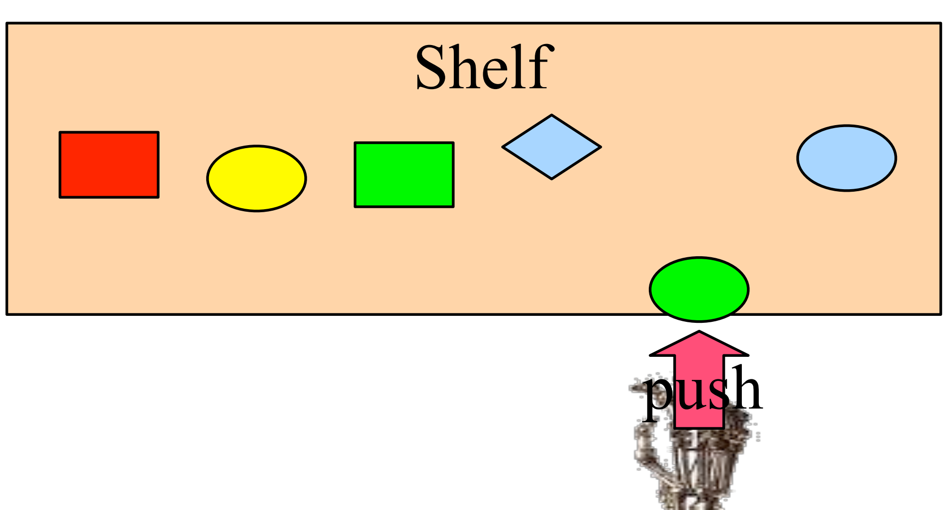
Inputs	Outputs	Function
(O, A)	E	Effect prediction
(O, E)	A	Action recognition/planning
(A, E)	O	Object recognition/selection

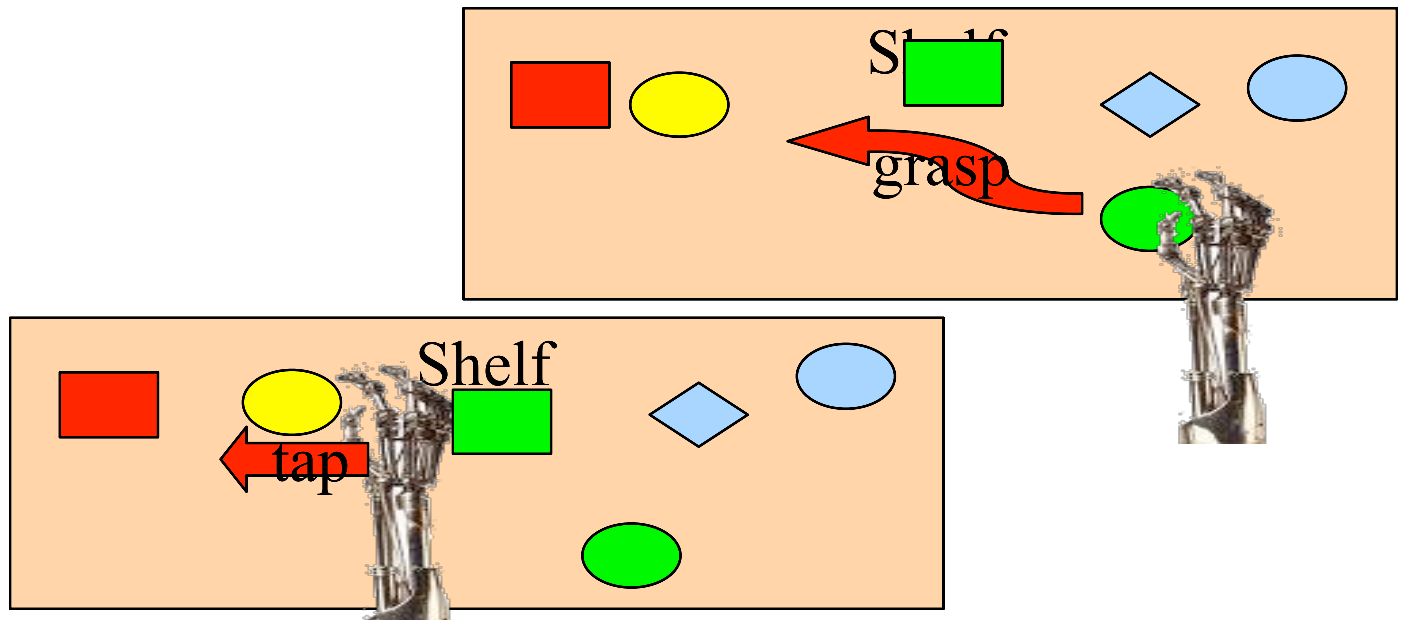


From two object interactions

Generalize to N

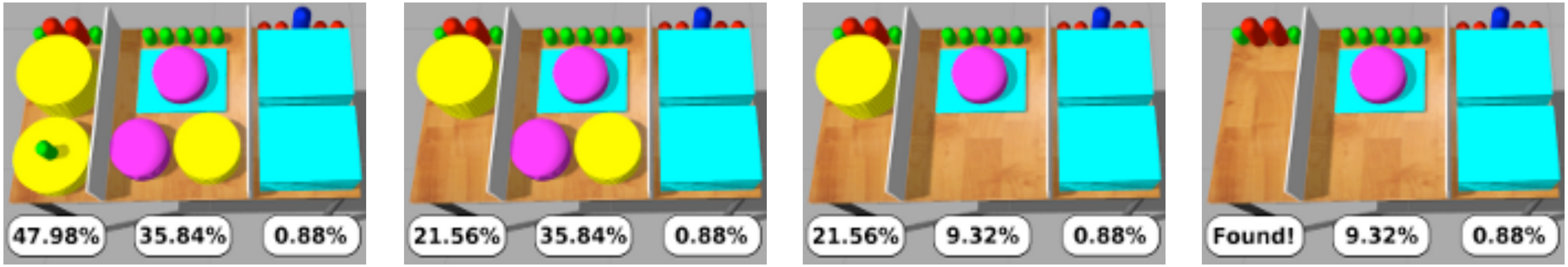
Moldovan et al. ICRA 12, 13, 14





Occluded Object Search

- Models of objects and their spatial arrangement
- different types of objects suitable for different tasks
- shelves with objects of different shape and size
- given a task, find an object to perform that task



[Moldovan et al ICRA 2014]

ProbLog for activity recognition from video

CAVIAR-INRIA human activity dataset

28 videos
≈ 26.500 frames

- Separation between low-level events (LLE) and high-level events (HLE)
  - LLE: *walking, running, active, inactive, abrupt*
  - HLE: *meeting, moving, fighting, leaving_object*
- Probabilistic Logic approach: *Event Calculus in ProbLog* (Prob-EC) to infer the high-level events from an **algebra** of low-level events.
- Example:

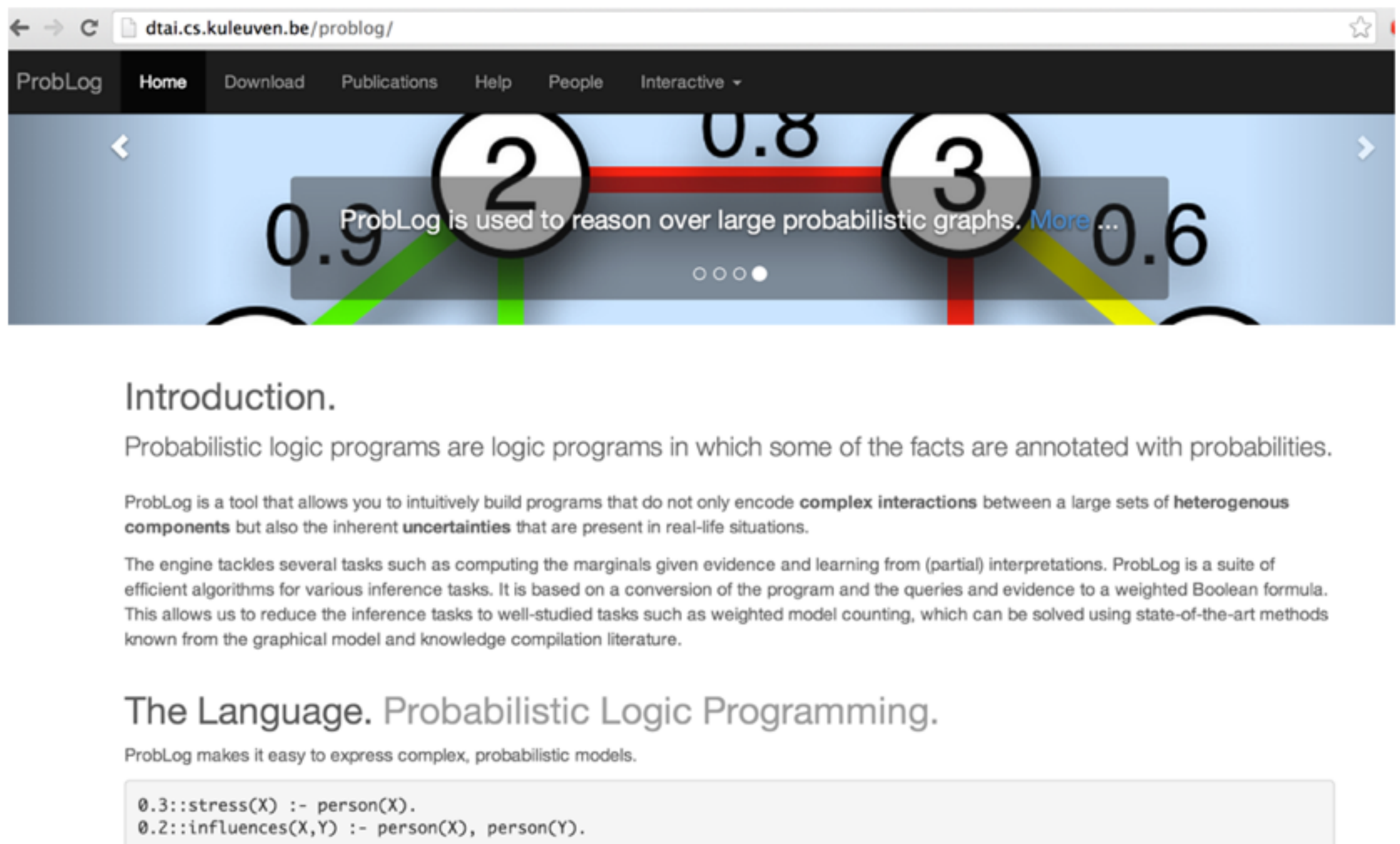
$$\begin{gathered} initiatedAt(fighting(P_1, P_2) = true, T) \leftarrow \\ happensAt(abrupt(P_1), T), \\ holdsAt(close(P_1, P_2, 44) = true, T), \\ not\ happensAt(inactive(P_2), T). \end{gathered}$$

[Skarlatidis et al, TPLP 13]

Vaishak Belle
Maurice Bruynooghe
Bart Demoen
Anton Dries
Daan Fierens
Jason Filippou
Bernd Gutmann
Manfred Jaeger
Gerda Janssens
Kristian Kersting
Angelika Kimmig
Theofrastos Mantadelis
Wannes Meert
Bogdan Moldovan
Siegfried Nijssen
Davide Nitti
Joris Renkens
Kate Revoredo
Ricardo Rocha
Vitor Santos Costa
Dimitar Shterionov
Ingo Thon
Hannu Toivonen
Guy Van den Broeck
Mathias Verbeke
Jonas Vlasselaer

Thanks !

http://dtai.cs.kuleuven.be/problog



PLP Systems

- **PRISM** http://sato-www.cs.titech.ac.jp/prism/
- **ProbLog2** http://dtai.cs.kuleuven.be/problog/
- Yap Prolog http://www.dcc.fc.up.pt/~vsc/Yap/ includes
  - **ProbLog1**
  - **cplint** https://sites.google.com/a/unife.it/ml/cplint
  - **CLP(BN)**
  - **LP2**
- **PITA** in XSB Prolog http://xsb.sourceforge.net/
- **AILog2** http://artint.info/code/ailog/ailog2.html
- **SLPs** http://stoics.org.uk/~nicos/sware/pepl
- **contdist** http://www.cs.sunysb.edu/~cram/contdist/
- **DC** https://code.google.com/p/distributional-clauses
- **WFOMC** http://dtai.cs.kuleuven.be/ml/systems/wfomc

References

Bach SH, Broecheler M, Getoor L, O'Leary DP (2012) Scaling MPE inference for constrained continuous Markov random fields with consensus optimization. In: Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS-12)

Broecheler M, Mihalkova L, Getoor L (2010) Probabilistic similarity logic. In: Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI-10)

Bryant RE (1986) Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers 35(8):677–691

Cohen SB, Simmons RJ, Smith NA (2008) Dynamic programming algorithms as products of weighted logic programs. In: Proceedings of the 24th International Conference on Logic Programming (ICLP-08)

Cussens J (2001) Parameter estimation in stochastic logic programs. Machine Learning 44(3):245–271

De Maeyer D, Renkens J, Cloots L, De Raedt L, Marchal K (2013) Phenetic: network-based interpretation of unstructured gene lists in e. coli. Molecular BioSystems 9(7):1594–1603

De Raedt L, Kimmig A (2013) Probabilistic programming concepts. CoRR abs/1312.4328

De Raedt L, Kimmig A, Toivonen H (2007) ProbLog: A probabilistic Prolog and its application in link discovery. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)

De Raedt L, Frasconi P, Kersting K, Muggleton S (eds) (2008) Probabilistic Inductive Logic Programming — Theory and Applications, Lecture Notes in Artificial Intelligence, vol 4911. Springer

Eisner J, Goldlust E, Smith N (2005) Compiling Comp Ling: Weighted dynamic programming and the Dyna language. In: Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-05)

Fierens D, Blockeel H, Bruynooghe M, Ramon J (2005) Logical Bayesian networks and their relation to other probabilistic logical models. In: Proceedings of the 15th International Conference on Inductive Logic Programming (ILP-05)

Fierens D, Van den Broeck G, Bruynooghe M, De Raedt L (2012) Constraints for probabilistic logic programming. In: Proceedings of the NIPS Probabilistic Programming Workshop

Fierens D, Van den Broeck G, Renkens J, Shterionov D, Gutmann B, Thon I, Janssens G, De Raedt L (2014) Inference and learning in probabilistic logic programs using weighted Boolean formulas. Theory and Practice of Logic Programming (TPLP) FirstView

Getoor L, Friedman N, Koller D, Pfeffer A, Taskar B (2007) Probabilistic relational models. In: Getoor L, Taskar B (eds) An Introduction to Statistical Relational Learning, MIT Press, pp 129–174

Goodman N, Mansinghka VK, Roy DM, Bonawitz K, Tenenbaum JB (2008) Church: a language for generative models. In: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI-08)

Gutmann B, Thon I, De Raedt L (2011a) Learning the parameters of probabilistic logic programs from interpretations. In: Proceedings of the 22nd European Conference on Machine Learning (ECML-11)

Gutmann B, Thon I, Kimmig A, Bruynooghe M, De Raedt L (2011b) The magic of logical inference in probabilistic programming. Theory and Practice of Logic Programming (TPLP) 11((4–5)):663–680

Huang B, Kimmig A, Getoor L, Golbeck J (2013) A flexible framework for probabilistic models of social trust. In: Proceedings of the International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction (SBP-13)

Jaeger M (2002) Relational Bayesian networks: A survey. Linköping Electronic Articles in Computer and Information Science 7(015)

Kersting K, Raedt LD (2001) Bayesian logic programs. CoRR cs.AI/0111058

Kimmig A, Van den Broeck G, De Raedt L (2011a) An algebraic Prolog for reasoning about possible worlds. In: Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-11)

Kimmig A, Demoen B, De Raedt L, Santos Costa V, Rocha R (2011b) On the implementation of the probabilistic logic programming language ProbLog. Theory and Practice of Logic Programming (TPLP) 11:235–262

Koller D, Pfeffer A (1998) Probabilistic frame-based systems. In: Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)

McCallum A, Schultz K, Singh S (2009) FACTORIE: Probabilistic programming via imperatively defined factor graphs. In: Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NIPS-09)

Milch B, Marthi B, Russell SJ, Sontag D, Ong DL, Kolobov A (2005) Blog: Probabilistic models with unknown objects. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05)

Moldovan B, De Raedt L (2014) Occluded object search by relational affordances. In: IEEE International Conference on Robotics and Automation (ICRA-14)

Moldovan B, Moreno P, van Otterlo M, Santos-Victor J, De Raedt L (2012) Learning relational affordance models for robots in multi-object manipulation tasks. In: IEEE International Conference on Robotics and Automation (ICRA-12)

Muggleton S (1996) Stochastic logic programs. In: De Raedt L (ed) Advances in Inductive Logic Programming, IOS Press, pp 254–264

Nitti D, De Laet T, De Raedt L (2013) A particle filter for hybrid relational domains. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-13)

Nitti D, De Laet T, De Raedt L (2014) Relational object tracking and learning. In: IEEE International Conference on Robotics and Automation (ICRA), June 2014

Pfeffer A (2001) IBAL: A probabilistic rational programming language. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)

Pfeffer A (2009) Figaro: An object-oriented probabilistic programming language. Tech. rep., Charles River Analytics

Poole D (2003) First-order probabilistic inference. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)

Richardson M, Domingos P (2006) Markov logic networks. Machine Learning 62(1-2):107–136

Santos Costa V, Page D, Cussens J (2008) CLP(BN): Constraint logic programming for probabilistic knowledge. In: De Raedt et al (2008), pp 156–188

Sato T (1995) A statistical learning method for logic programs with distribution semantics. In: Proceedings of the 12th International Conference on Logic Programming (ICLP-95)

Sato T, Kameya Y (2001) Parameter learning of logic programs for symbolic-statistical modeling. J Artif Intell Res (JAIR) 15:391–454

Sato T, Kameya Y (2008) New advances in logic-based probabilistic modeling by prism. In: Probabilistic Inductive Logic Programming, pp 118–155

Skarlatidis A, Artikis A, Filiopou J, Paliouras G (2014) A probabilistic logic programming event calculus. Theory and Practice of Logic Programming (TPLP) FirstView

Suciu D, Olteanu D, Ré C, Koch C (2011) Probabilistic Databases. Synthesis Lectures on Data Management, Morgan & Claypool Publishers

Taskar B, Abbeel P, Koller D (2002) Discriminative probabilistic models for relational data. In: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-02)

Thon I, Landwehr N, De Raedt L (2008) A simple model for sequences of relational state descriptions. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML/PKDD-08)

Thon I, Landwehr N, De Raedt L (2011) Stochastic relational processes: Efficient inference and applications. Machine Learning 82(2):239–272

Van den Broeck G, Thon I, van Otterlo M, De Raedt L (2010) DTProbLog: A decision-theoretic probabilistic Prolog. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI-10)

Van den Broeck G, Taghipour N, Meert W, Davis J, De Raedt L (2011) Lifted probabilistic inference by first-order knowledge compilation. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)

Vennekens J, Verbaeten S, Bruynooghe M (2004) Logic programs with annotated disjunctions. In: Proceedings of the 20th International Conference on Logic Programming (ICLP-04)

Vennekens J, Denecker M, Bruynooghe M (2009) CP-logic: A language of causal probabilistic events and its relation to logic programming. Theory and Practice of Logic Programming (TPLP) 9(3):245–308

Wang WY, Mazaitis K, Cohen WW (2013) Programming with personalized pagerank: a locally groundable first-order probabilistic logic. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM-13)